WAP E-MAIL SYSTEM (WAPES)

Submitted By:

TAN WANG WEE
WEK 990238

Submission Date

25 JANUARY 2002

Supervisor

PN. NORIZAN MOHD YASIN

Moderator

EN. WOO CHAW SENG

ABSTRACT

This project is to build an application based on WAP (Wireless Application Protocol), where users could check their e-mails via their WAP phone. The objective of this project is to provide mobile users with the facility to check their e-mails while on the move. As such, users would be required to create a login account on the e-mail service created, which is accessible via the World Wide Web or through WAP phones.

This project is divided into two modules - the Web Module and WAP Module. The Web Module is an e-mail service site, allowing users to sign up for e-mail accounts, or for existing users to login to their accounts. Users would be able to perform typical e-mail functions on this module.

On the other hand, the WAP Module allows WAP users to use their WAP browsers on their mobile phones to check their e-mails. They could also perform typical e-mail functions via their WAP phones. This means that mobile users do not need to look for Internet connected PCs to check their important e-mails. They could simply check their e-mails via their WAP phones at anytime and anywhere according to their convenience.

ACKNOWLEDGEMENT

The development of the WAPES is proposed through the research, advice and assistance of many individuals. First and foremost, I would like to extend my most sincere gratitude to Pn. Norizan, my project supervisor who is providing me unlimited support and guidance all the while.

Utmost gratitude to my fellow course mates for sharing their knowledge throughout the duration of this proposal. And to my dearest family, who has always been encouraging and understanding throughout this project.

Lastly, I would like to thank all the people who have directly or indirectly assisted me to make this proposal a very meaningful and memorable experience to me.

CONTENT

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

Over recent years, the trend toward smaller and faster devices, coupled with the need for information access on the move, has paved the way for a new technology that brings together the two worlds of the Web and the mobile phone. The **Wireless Application Protocol (WAP)** is an industry-wide standard, defining a communications protocol and application environment that allows access to Internet content and services from mobile phones. Designed from the ground up for low-power, small-screen devices with limited input capabilities and low bandwidth, WAP has enabled the development of a new breed of Internet-ready phones at mass-market prices.

In 1997, Phone.com got together with Ericsson, Motorola and Nokia to form the **WAP Forum**, with the aim of creating a standardized solution to the problem of how to provide Internet access from mobile phones. In 1999, the first WAP phones hit the streets.

The number of mobile phones in the world is increasing every day at an astonishing speed, with analysts forecasting that there will be more than a billion mobile phones in use within the next five years [1], and that over half of Internet access will be through non-PCs. The mobile phone has become a part of daily life for many people, and together with a watch, is the only electronic device that many people carry around everywhere with them, all day long. Therefore, it is undeniable that WAP would play an important role in every aspect of future telecommunications industry.

1.1 Project Introduction

WAP and E-mail

E-mail and WAP are fast becoming the most demanded combinations of technology by both corporations and general consumers. The popularity achieved by the very limited SMS technology indicates that demand for messaging via mobile phones certainly exists, and giving mobile phones all the functionality of e-mail seems to be the next logical step.

E-mail is a substantially more advanced technology than SMS, even if it is only used for simple SMS-like text messages. Message recipients are not limited on how they receive the message when using e-mail. Rather than being able to access the message from a single mobile phone, the user can choose to access it from whatever client e-mail software he or she prefers, whether that is another WAP phone, a home PC, laptop, or even a UNIX workstation. E-mail unlike SMS, allows for the recipient to have an address that is more like 'natural language' than a phone number, and is thus easier to remember. Furthermore, e-mail provides the ability to mail 'group' addresses; for example all@wapbook.org. E-mail also has substantial multimedia functionality, and can use a variety of security protocols. [1]

1.2 Project Motivation

WAP devices and e-mail capabilities allow for a useful synergy of personal communication technology: delivering the convenience of portability from mobile phones, whilst allowing instant access to e-mail, providing asynchronous access to written messages.

E-mail is a substantially more advanced technology than SMS. With SMS, users are limited by only 160 chunks of text. However, with e-mail, message recipients are not limited in the following ways: [1]

- In how they receive the message
- Not only being able to access message from a single mobile phone, but, choose to access it from whatever client e-mail software, whether that is another WAP phone, a home PC, laptop, or even a UNIX workstation.
- E-mail, unlike SMS, allows for the recipient to have an address that is more like 'natural-language' than a phone number
- E-mail provides the ability to mail 'group' addresses

Some corresponding figures on mobile and WAP usage are listed below in order to assess the potential market that WAP e-mail functionality may reach (source: Durlacher Research Ltd):

- There are currently 300 million mobile subscribers, growing at 50% per annum (PC growth globally is now only about 20% p.a. and falling)

- WAP penetration into the mobile phone market is predicted to be 8% in 2000, 22% in 2001, 50% in 2002 and 85% in 2003
- By 2003, over 50% of Internet access will be by non-PC devices (Meta Group)
- By 2005, 1 billion mobile devices will be used worldwide (Gartner Group)

1.3 Project Objective

The objectives of the WAPES project are as follows:

- To enable WAP users perform standard e-mail functionality via WAP devices
- To enable users check e-mails not only from the Internet, but also from WAP devices anytime and anywhere

1.4 Project Scope & Limitation

The scope of the project means the coverage of the project as in what are the limitations of the application and what are the boundaries set upon the application.

- This project would be designed for use by people who needs to check e-mails when they do not have access to an Internet linked computer
- In the actual environment, WAP devices, i.e. a WAP phone and a WAP gateway, are the tools needed. However, for this project, a simulator would be used instead

to simulate the WAP application. In other words, the simulator functions as a substitute to the WAP phone and gateway, by enabling direct access to the web server and encoding the WML.

- The simulation has to be done in Windows based platform because most of the useful simulators can only be run in Windows based platform.

- Limitations of WAP devices – less powerful CPUs, less memory (ROM, RAM), smaller displays, different input devices

- The speed of data transmission rates of 9.6 kilobits per second is slow compared to the fixed line connections.

- The cost of using e-mail via WAP phone is high. For a normal dial-up connection, the cost is only RM 0.025/minute all day. However, the following table shows the mobile access fees for WAP:

Peak rate (9am – 7pm)	RM 0.30/minute
Off peak rate (7pm – 12 am)	RM 0.15/minute
Super off-peak rate (12am – 9am)	RM 0.025/minute

1.5 Project Schedule

The project starts in June 2001 and will be expected to complete by January 2002.

ID	Task Name	2001							2002	
		Jun	July	Aug	Sept	Oct	Nov	Dec	Jan	Feb
1	System Initiation									
2	Literature Review									
3	Requirements Analysis									
5	System Design									
6	System Building									
7	System Implementation									
8	Documentation									

1.6 Conclusion

As the project motivation, objectives, scope and limitations were discussed in detail, we shall proceed to Chapter 2, where detailed literature review is being done.

CHAPTER 2: LITERATURE REVIEW

2.1 Importance of Literature Review

The purpose of a literature review is to sufficiently equip the developers with some knowledge of the strengths and limitations of several development tools available. This is to enable the developer to choose the right tool to use to develop the system.

It provides a wide review on the materials being used for the system. At this stage, all kind of materials such as software, development tools, web server, simulator, and dynamic scripting languages are being researched to compare their advantages and disadvantages. From this, the most suitable choices will be taken to ensure the successful implementation of this project.

Most importantly, it also provides reviews on existing systems. These existing systems are further analyzed and act as a reference point for the project. Some important features that need to be further improved are documented.

2.2 Findings

2.2.1 Wireless Application Protocol (WAP)[11]

WAP stands for **Wireless Application Protocol**, and is a set of specifications for developing web-like applications that run over wireless networks. The basic aim of WAP is to provide a web-like experience on small portable devices - like mobile phones and personal digital assistants (PDA's).

2.2.1.1 Technical Background

WAP protocols are open-specification, meaning that they are discussed, designed, and owned by a democratic consortium of interested parties - the WAP Forum. This is a bit like the W3C group that handles specifications for HTML and so on.

The WAP protocols are analogous to many of those used in existing Internet technology, yet are optimized for the challenges raised by small, narrowband client devices (such as mobile phones) and limited over-the-air bandwidth.

The WAP Protocols cover both the application level (WML and WMLScript; collectively known as **WAE**, the Wireless Application Environment), and the underlying transport layers (**WSP** and **WTP**, **WTLS**, and **WDP**), as depicted in Figure 2-2.

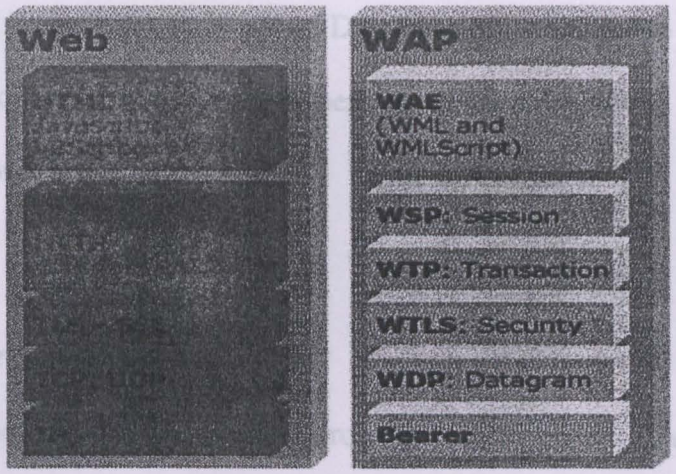


Figure 2-2: WAP stack[11]

The WAP stack illustrated in Figure 1, has 5 different layers:

- **Application Layer:** WAE (Wireless Application Environment) provides an application environment intended for the development and execution of portable applications and services.
- **Session Layer:** WSP (Wireless Session Protocol) supplies methods for the organized exchange of content between client/server applications.
- **Transaction Layer:** WTP (Wireless Transaction Protocol) provides different methods for performing transactions, to a varying degree of reliability.
- **Security Layer:** WTLS (Wireless Transport Layer Security) is an optional layer that provides, when present, authentication, privacy and secure connections between applications.

- **Transport Layer:** WDP (Wireless Datagram Protocol) is the bottom layer of the WAP stack, which shelters the upper layers from the bearer services offered by the operator.

2.2.1.2 The Servers

The Internet comprises clients (browsers) and servers (web servers), which actually host the pages to view - or the applications that produce the pages on-the-fly.

WAP is no different. Clients are now the mobile devices, and the servers are still hosting static pages and applications.

The only consideration to now make is to connect the mobile network to the Internet - so that the phone can make a request for the page and it will be able to reach the server. (And of course the response has to make it back).

To connect these two mega-networks, the WAP Specification assumes there will be a WAP Gateway. At its simplest level, this is a 'stack converter'. It converts the 'WAP' request into a 'Web' request and the 'Web' response into a 'WAP' response.

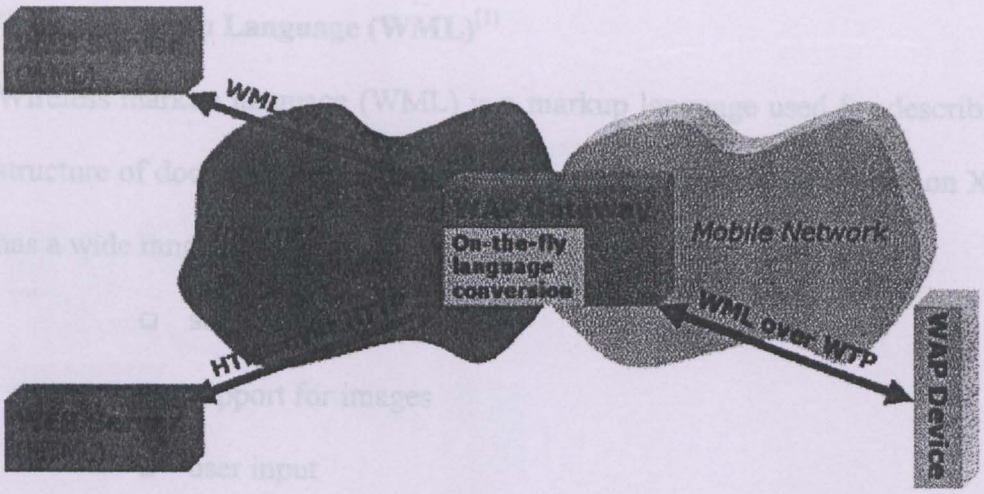


Figure 2-3: WAP Gateway[11]

2.2.1.3 WAP Server

A WAP server is simply a combined web server and WAP gateway. The purpose of combining these is security. When accessing a secure service (using WTLS) the data is sent encrypted to the WAP gateway. The gateway decrypts it and encrypts it using SSL before passing it on to the web server. When the data returns, it will be decrypted and encrypted using WTLS before being sent to the WAP device. It's easy to see where the weak link is: the WAP gateway. However, by using a WAP server, this weak link is not only eliminated, the WAP server also controls the content and the gateway where the encryption/decryption process takes place.

2.2.2 Wireless Markup Language (WML)[1]

Wireless markup language (WML) is a markup language used for describing the structure of documents to be delivered to wireless devices. It is based on XML. It has a wide range of features:

- support for text
- support for images
- user input
- variables
- navigation and history stack
- internal support
- optimization for narrow-band

2.2.3 WMLScript[1]

WMLScript is a lightweight procedural scripting language, which is based on ECMAScript, the standardized version of JavaScript. It adds intelligence to the client, providing a set of libraries for mathematical operations, string manipulation, etc., collaborates with WML, and helps prevent unnecessary connections with the server.

2.2.4 E-mail System[1]

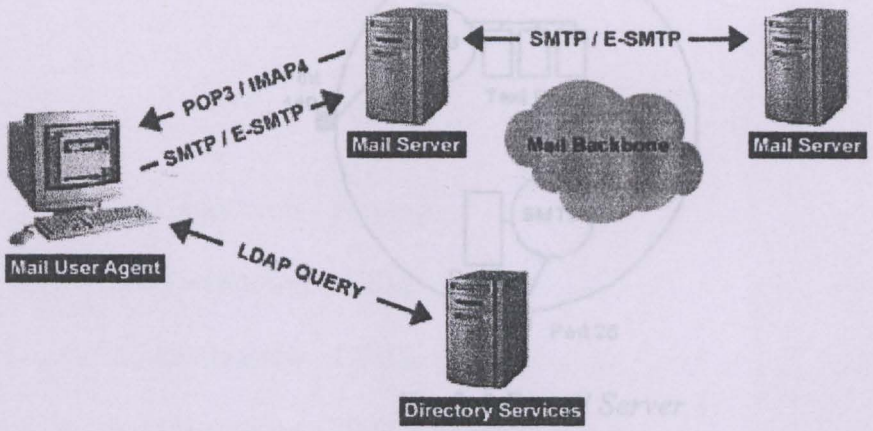


Fig 2-4 A Typical E-mail System [1]

In a typical e-mail system, an **e-mail client** or **Mail User Agent (MUA)** is used to retrieve e-mails and to perform all standard e-mail functionalities. Many people use well-known stand-alone clients like Microsoft Outlook, Outlook Express, Eudora or Pegasus. People who subscribe to free e-mail services like Hotmail or Yahoo use an e-mail client that appears in a Web page.

Given that an e-mail client is working on a machine, users are ready to send and receive e-mail. All that is needed is an **e-mail server** for the client to connect to.

An e-mail system consists of two different servers running on a server machine. One is called the **SMTP Server**, where SMTP stands for Simple Mail Transfer Protocol. The SMTP server handles outgoing mail. The other is a **POP3 Server**, where POP stands for Post Office Protocol. The POP3 server handles incoming mail. The SMTP server listens on well-known port number 25, while POP3 listens on port 110. A typical e-mail server looks like this:

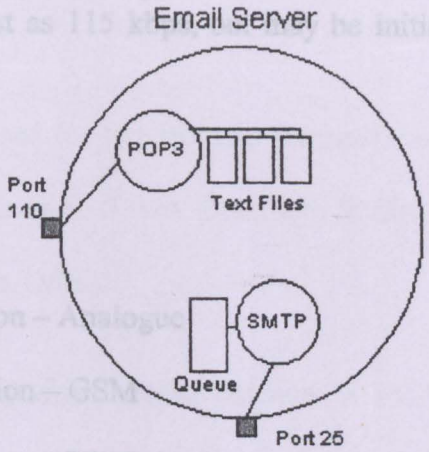


Fig 2-5 E-mail Server

To summarize, from a user's point of view e-mail is sent via SMTP, collected from their mailbox using POP3 or IMAP, and any address book information is searched for using LDAP (or ACAP).

2.2.5 Current Terms of Mobile Communication[9]

- **GSM**

  Global System for Mobile communication. The de facto digital cellular radio network in Europe. (Includes GSM 900, GSM 1800, GSM 1900.)[5]

- **GPRS**

  General Packet Radio Service - A mobile data communications service that is widely expected to be the next major step in the evolution of TDMA and GSM. It will be a packet-based system that could be used for "bursty" data applications such as mobile Internet browsing, e-mail and push technologies. GPRS has been

demonstrated as fast as 115 kbps, but may be initially deployed with speeds of 28.8 kbps.[5]

- **3G**
  - 1st Generation – Analogue
  - 2nd Generation – GSM
  - 2.5 Generation – GPRS
  - 3rd Generation - UMTS IP based network

- **SMS**

SMS is an addition to the GSM Standard that enables text messages of up to 160 characters on GSM networks and 190 characters on some other networks to be sent between mobile phones.

SMS has been around just about as long as GSM but it's popularity has only really come to the fore in the past few years with the youth market in Scandanavian countries taking the lead. SMS is popular because it is cheap (about 10 pence/message), quick to type and fun to receive.

Besides messaging between mobile phones SMS can be incorporated into web and traditional applications to alert users of events such as a new email arriving or a share price movement. SMS is used in conjunction with other technologies such as WAP to provide the bearer service for PUSH messages.

- **Bluetooth**

Bluetooth is a standard for the wireless transmission of data between devices by the use of short range radio waves. Bluetooth Radios operate in a band at 2.4GHz. The gross data rate is 1Mb/s.

Possible uses are synchronisation between a PDA and desktop computer or internet connectivity to a laptop through a mobile phone, all without the need for any wires, docking stations or infrared ports.

Bluetooth got its name from the 10th Century Viking King - Harald Bluetooth

- **I-Mode**

i-Mode is a wireless technology developed by the Japanese company NTT DoCoMo that enables users to access Internet services via their cellular phones. I-mode (the I stands for information) is based on packet data transmission technology. This means that i-Mode is always online, and therefore users are charged only for how much information they retrieve, not how many minutes they are using it for. i-Mode can be used to exchange e-mail with computers, personal digital assistants (PDAs) and other i-Mode cellular phones.

As the Japanese giant NTT DoCoMo recently expanded their services into the European market we may witness a convergence of the two global standards in Mobile Internet technology, WAP and i-Mode. They are essentially similar

technologies allowing the user to access web based information with transactional capabilities from a mobile phone. However there are some major differences between the i-Mode and WAP.

The difference between WAP and i-Mode:

i) **programming language used**

WAP uses the markup language WML (Wireless Markup Language) while I-Mode uses CHTML (Compact HTML).

Compact HTML has an advantage over WML in that a large majority of WML developers come from the "web" world where they are used to HTML. However, the future of internet content serving is XML, and from XML the step to WML is hardly noticeable. It's much more noticeable with Compact HTML or HTML.

ii) **I-Mode is an always-on connection**

This means that you do not have to 'dial up' to access a site and email is instantly sent to your phone. This uses GPRS (General Packet Radio Service) technology and will be made available to us some time next year.

Email will become as instant as SMS, and as phone technology advances, we will be able to send, receive and store large files on our wireless devices.

iii) **billing**.

As I-Mode is based on a packet-data (9600bps) transmission system, subscribers will be charged according to the volume of data transmitted, not the time spent on line. With the 'always on' connection this method is necessary as the time spent is unlimited.

While WAP and i-Mode both have different specifications, WAP specifications are being created so as to take in the i-Mode specifications. The WAP specification has been updated to version 2 in order to become more like it's Asian counterpart. This is overseen by the WAP forum, which is the industry association involving companies form every sector of the wireless industry value chain.

Also, for now, i-Mode is being used mostly in Japan, although NTT DoCoMo (who have 20 million subscribers) hope to move i-Mode service into Britain and the rest of Europe in the coming future. In doing so, NTT DoCoMo has begun to provide English content on their i-Mode cellular phones for foreigners living in Japan.

2.3 Tools Considered

2.3.1 WAP Development Toolkit[11]

Creating, testing and demonstrating WAP applications is made easy with the numerous Development Toolkits available for downloads. Utilizing an easy-to-use PC environment, the Toolkits can simulate WAP content on compatible devices.

SDK Provider	SDK	Emulator	Editor	Debugging Tools	Sample Apps	Online Docs
ThinAirApps	ThinAir Wireless SDK	Yes	Yes	Yes	Yes	Yes
MobileDev	MobileDev		Yes	Yes	Yes	Yes
Wapalize	Wapalize WAP Development Tool Kit	Yes	Yes		Yes	Yes
Nokia	Nokia WAP Toolkit	Yes	Yes	Yes	Yes	Yes
Motorola	Mobile Application Development Kit (ADK)	Yes	Yes	Yes	Yes	Yes
Ericsson	WAPIDE SDK 2.1	Yes	Yes			Yes
Phone.com	UP.SDK 4.0	Yes			Yes	Yes
WAPObjects	WAPObjects		Yes	Yes	Yes	Yes
WAPMine	WAPPage 1.0		Yes		Yes	Yes
Perfect Solutions	CardONE	Yes	Yes			
Dynamical Systems Research	WAP Developer Toolkit 1.0	Yes	Yes	Yes	Yes	Yes
PWOT	PWOT WML-Tools	Yes		Yes		Yes

2.3.2 Development Platform

2.3.2.1 Windows 2000 Professional

Windows 2000 Professional is the Windows operating system for business desktop and laptop systems. It is used to run software applications, connect to Internet and intranet sites, and access files, printers, and network resources.

Built on Windows NT® technology and the easy-to-use, familiar Windows® 98 user interface, Windows 2000 Professional gives users increased flexibility. The integrated Web capabilities connect to the Internet from anywhere, at anytime. In addition, broad peripheral and mobile computer support make Windows 2000 Professional an ideal operating system for a workforce that increasingly relies on notebook computers. Further, it's reliability and manageability enhancements make desktop management simpler and more efficient.

Windows 2000 Professional is reliable, great for mobile users, easy to use and maintain, and ready for the Internet.

2.3.2.2 Unix

Developed by Ken Thompson in 1969, the most popular general-purpose, multi-user operating system in the world. The coauthor is Dennis Ritchie, who created the C programming language. Unix became the first source-portable operating system in 1974 when it was implemented in C. Unix is the subject of an international standardization effort. It is a 32-bit system with excellent multitasking capabilities. Similar systems include Open Software Foundation (OSF), Version 7, Berkeley Software Distribution (BSD), Xenix, Ultrix, Linux, and GNU. *Unix* is a trademark, owned originally by AT&T, sold to Novell in 1993, and subsequently to Santa Cruz Operation, Inc. (SCO) in 1996. In 1993, Novell assigned the rights to the Unix name to X/Open Company Ltd., and they license it worldwide.

Unix possess the following unique features:

- Multi-users – Unix can support one to hundreds of users with users concurrently running different sets of program.
- Multi-tasking – User can run more than one job at a time.
- Kernel – The kernel is responsible for controlling the computer's resources and scheduling user jobs so that each user gets their fair share including CPU, disk storage, printers and tape drivers.
- Security – Unix system allows users to protect their data from being accessed by other users. Users can share selected data with an effective protection scheme.

2.3.2.3 Linux

Short for Linus Unix; an implementation of the Unix kernel that runs on Intel and Alpha hardware in the general release. Versions of Linux for SPARC, PowerPC, and SGI are in active development. Versions that are ported to microcomputers support networking and shells. Linus Torvalds, the primary copyright holder, coordinates development of the Linux kernel. The whole kernel is available under the GNU general public license.

Linux possess the following features:

- Utilities – Linux can serve as a server because it has a variety of supporting tools such as Apache web server and can support Oracle and SQL database
- Stability and Reliability – Linux is a very stable operating system which means that all programs operating on it would not fail
- Security – Linux also supports multi-user and each of the users can only access to the system through password, user's resources are also segregated accordingly by the administrator.

2.3.3 Web Server

The Internet utilizes two types of computers -- servers and clients. Servers are the computers that host the pages that we view. The clients are our PCs, or the computers that we view these pages from.

WAP technology works much in the same way. The difference is that, instead of accessing the Internet from our PCs, we are using mobile phones. The mobile phones become our clients. Servers are still hosting the pages that we view.

2.3.3.1 Apache Web Server[4]

Apache is an ongoing group software development effort. Apache servers dominate the Web in numbers, in part, because it is free and performs very efficiently - it is powerful enough that IBM has licenced it for its own WebSphere application server package. In the period from 1996 through 1999, Apache has enjoyed the highest increase in Internet Web sites of all Web servers, according to a Netcraft survey. Currently, Apache is more widely used than all the other Web servers combined.

Apache runs on many operating systems, (AIX, BSD/OS, FreeBSD, HP-UX, Irix, Linux, Microsoft NT, QNS, SCO, and Solaris) and the hardware that supports them. Apache has a built-in search engine and HTML authoring tools and supports FTP. Apache can be managed from a server console or a Web browser. Apache's security is well thought out, with support for password authentication

and digital certificate authentication. Access can be restricted by domain name, by IP address, or by user and group. Apache can prohibit access by directory or file, and supports SSL. Apache also supports Active Server Pages (ASPs) and Java Servlets. It also supports the ODBC standard and can access Oracle, Sybase, Microsoft SQL Server, and IBM's DB2 databases.

2.3.3.2 Microsoft Internet Information Server[4]

Microsoft's Internet Information Server (IIS) comes bundled (free) with Microsoft's Windows NT operating system. IIS serves equally well as an Intranet Web server or a public web server program. IIS is the second most installed Web server software for public and intranet Web sites. A robust and capable Web server program, IIS is suitable for small sites right up to enterprise-class sites doing high transaction volumes.

Currently, IIS runs only on the Windows NT operating system. IIS includes and integrated search engine that allows users to create custom search forms with a variety of tools, including ASP, ActiveX Data Objects, and SQL database queries. The IIS Web server software also includes Microsoft's Front Page HTML development tool and reporting tools from Crystal reports. IIS supports FTP, allowing users to download files and data from the IIS server site with the FTP protocol. Security in IIS is tightly integrated with Windows NT's operating system security. Thus, NT basic access control mechanisms (username/password) and SSL software encryption are also provided in IIS. IIS includes a built-in

certificate server that allows organizations to issue and manage digital certificates verifying identities. Database support includes ODBC and Microsoft SQL.

2.3.3.3 Microsoft Personal Web Server

Personal Web Server (PWS) is similar to any other Web server, except that it runs on Windows 95. However, Windows 95 isn't as robust of a networking platform as Windows NT. Therefore, PWS may not perform as well as a Web server running on a Windows NT Server.

Web Server is actually made up of two separate components: the Web server and an FTP server. Both of these components function independently of one other. We have the option of running just a Web server, just an FTP server, or both.

It provides most of the functionality of Microsoft Internet Information Server, including the ability to:

- Publish Web pages on the Internet or over a LAN on an intranet.
- Support Microsoft ActiveX programs.
- Transmit or receive files by using the FTP service.

2.3.4 Dynamic Scripting Language

2.3.4.1 JavaServer Pages (JSP)

JavaServer Pages (JSP) technology allows web developers and designers to rapidly develop and easily maintain, information-rich, dynamic web pages that leverage existing business systems. As part of the Java family, JSP technology enables rapid development of web-based applications that are platform independent. JavaServer Pages technology separates the user interface from content generation enabling designers to change the overall page layout without altering the underlying dynamic content.

JavaServer Pages technology uses XML-like tags and scriptlets written in the Java programming language to encapsulate the logic that generates the content for the page. Additionally, the application logic can reside in server-based resources (such as JavaBeans component architecture) that the page accesses with these tags and scriptlets. Any and all formatting (HTML or XML) tags are passed directly back to the response page. By separating the page logic from its design and display and supporting a reusable component-based design, JSP technology makes it faster and easier than ever to build web-based applications.

JavaServer Pages technology is an extension of the Java Servlet technology. Servlets are platform-independent, 100% pure Java server-side modules that fit seamlessly into a web server framework and can be used to extend the capabilities

of a web server with minimal overhead, maintenance, and support. Unlike other scripting languages, servlets involve no platform-specific consideration or modifications; they are Java application components that are downloaded, on demand, to the part of the system that needs them. Together, JSP technology and servlets provide an attractive alternative to other types of dynamic web scripting/programming that offers platform independence, enhanced performance, separation of logic from display, ease of administration, extensibility into the enterprise and most importantly, ease of use.

The JSP specification is the product of industry-wide collaboration with industry leaders in the enterprise software and tools markets, led by Sun Microsystems. Sun has made the JSP specification freely available to the development community, with the goal that every web server and application server will support the JSP interface. JSP pages share the "Write Once, Run Anywhere" characteristics of Java technology. JSP technology is a key component in the Java 2 Platform, Enterprise Edition, Sun's highly scalable architecture for enterprise applications.

2.3.4.2 Active Server Pages (ASP)

ASPs are server-generated pages which can call other programs to do things like access databases, serve different pages to different browsers - basically, anything we used to do with CGI. ASP is almost as efficient as writing code directly to the server's application program interface, and it's a lot more efficient than CGI because it runs as a service and can take advantage of multithreaded architectures.

Here's the official word from the Microsoft site: "Active Server Pages is an open, compile-free application environment in which you can combine HTML, scripts, and reusable ActiveX server components to create dynamic and powerful Web-based business solutions. Active Server Pages enables server-side scripting for IIS with native support for both VBScript and Jscript."

ASP was introduced so that all the Visual Basic programmers in the world would feel needed again, now that the Web had shoved Unix geeks back into the limelight (they tried and failed with VBScript, which pretty much died for a whole host of reasons).

ASP evolved into an "open technology framework," meaning do not need to use Microsoft products to create code in it, though that's still the best way to go. Nowadays, ASP pages can be created using whatever language, but VBScript is still the most common choice. It seems likely that more people will choose to use ASP. ASPs can also take advantage of COM and DCOM (Component Object Model and Distributed Component Object Model) objects with minimum effort.

2.3.5 Application Programming Interfaces (APIs)[1]

2.3.5.1 Microsoft's Collaboration Data Objects (CDO)

CDO is an extremely simple-to-use API that is based on COM objects. It's the API to use if coding in Visual Basic, Visual C++, or scripting languages on Microsoft platforms which will be the case if using ASP to generate WML pages dynamically. CDO comes in two variants, CDO and CDONTS (CDO for NTServer).

CDO is designed to run against Microsoft's Exchange Server a sophisticated mail server that can send and receive mail and organize all users' mail accounts for a large organization. Exchange Server's functionality is considerable and is accessible practically in its entirety from CDO. That means that although CDO is still relatively simple to use, it is more complex than for very basic send/receive email functionality. For this reason Microsoft developed CDONTS as an alternative. CDONTS can run either against Exchange Server or against the simple SMTP service that is shipped with Internet Information Server (IIS). CDONTS can't do much beyond sending and receiving e-mails, but that means it is incredibly easy to use CDONTS to programmatically send an email with just three lines of VBScript code.

Using CDO, complex messaging functionality can be pulled together with other related COM based programming models such as ActiveX Data Objects (ADO)

and the Active Directory Service Interface (ADSI). This linkage will provide mechanisms to manipulate mail, and easily extend CDO applications, to include other areas of personal information management functionality, such as contact and calendar management.

2.3.5.2 JavaMail

Released by Sun in August 1998, JavaMail aims to revolutionize access to mail systems in the way that Java Database Connectivity (JDBC) revolutionized access to databases. The advantages of using JavaMail include:

- It is available on the majority of current operating systems
- It offers an e-mail API that is both flexible and easy to use
- It offers excellent networking capabilities as standard

In the public imagination, Java and the Internet are inextricably linked, and to the programmer Java continues to offer the simplest and most elegant language with which to implement network-centered applications. The wealth of networking APIs and server-side solutions offered by Java makes it an obvious choice for implementing the distributed backend systems needed for our WAP applications. We can make use of Java on the server-side to effectively scale and efficiently partition our WAP applications.

2.3.6 Mail Server

2.3.6.1 Overview

For Internet mail servers, a very important factor is the support of standards. The major protocols are SMTP (Simple Mail Transfer Protocol) for outgoing mail and POP3 (Post Office Protocol) for incoming mail. A more recent protocol, which has yet to make the inroads expected of it, is IMAP4 (Internet Messaging Access Protocol). IMAP offers a number of important features, including user management of mail on the server. Other Internet protocols include ESMTP (Extended Simple Mail Transfer Protocol), APOP (Authenticated Post Office Protocol), MIME (Multipurpose Internet Mail Extensions), and Ph (Directory Access protocol). Many mail servers are also adding S/MIME, SSL, or RSA support for message encryption; and LDAP (Lightweight Directory Access Protocol) support to access operating system directory information about mail users. In general, the more standards a server supports, the better.

Because of the ever-increasing volume of e-mail traffic, the scalability of an e-mail server -- even for a relatively small organization -- can be a major issue. It is also important because mail server products tend to differentiate themselves by the message volume of their target customer and the number of e-mail accounts they support. Some are intended for corporate and enterprise e-mail, often processing millions of messages in a short period of time and providing accounts for more than 100,000 users; others target small businesses that may handle only a few thousand messages a week and support less than 100 users.

The highly publicized viruses that attack through e-mail clients have put the spotlight on e-mail as a vulnerable point in an enterprise's firewall. In response, mail server vendors (along with major client vendors such as Microsoft) have begun producing add-ons and built-in features that will help to scan mail, segregate questionable messages, and deal with viruses and spam. As might be expected, this is a rapidly expanding element of e-mail servers, and should be considered important when comparing products.

E-mail servers rank high in difficulty to install and manage. That's because they are tied to an inherently variable source-Internet connection and mail traffic -- and because they require constant attention to user lists, user rights, and message storage. Thus, it's important to look for servers that provide a certain modicum of ease of use. A GUI interface is nice, but it is not necessarily the route to easy administration. The type of interface is often related to platform (operating system), and it is important to keep in mind that many products are intended to run on a single platform (e.g., Unix or Windows).

2.3.6.2 Commercial Software

There are many commercial mail servers available. Some examples are listed below:

- 602Pro LAN Suite
- Avirt Mail Server
- CommuniGate Pro
- Dmail Email Server
- Emurl
- Eudora Internet Mail Server
- Imail Server
- MailMax
- Microsoft Exchange Server
- NTMail
- VOPMail

Microsoft Exchange Server 2000

- An enterprise-level mail and groupware server for Windows NT platform.
- It is a commercial software that costs about $995+, free 120-day evaluation.
- Created to take full advantage of the capabilities of Windows 2000
- Offers integrated management of networking and messaging infrastructure. For rapid application development, Exchange 2000 delivers built-in services such as calendaring, contact and task management, discussion groups, and document-centric workflow as well as support for Web-standard protocols, including Extensible Markup Language (XML) and Hypertext Transfer Protocol (HTTP).

Imail Server

- Full-featured low-cost mail server with some groupware functionality[16].
- It is a standards-based e-mail server system for Microsoft Windows NT/2000.
- It supports any POP3, IMAP4 mail client or, through Web messaging, any Web browser.
- IMail Server 7.0 is targeted toward ISPs, ASPs, and small to midsize enterprises. It comes in three classes of licenses: a 250 user license for $995, a 1,000 user license for $1,695, and a license for an unlimited number of users priced at $2,495.

2.3.6.3 Freeware

As can be seen, the above mail servers are commercial software that costs money. As such, for the purposes of the WAPES project, mail server freeware were looked out for.

1st Class Mail Server *(by 1CIS.com)*

- Free mail server with unlimited domains and mailboxes.
- It is fully compatible with SMTP/POP3 protocols, and it also supports ATRN for virtually hosted domains.
- The program works with any kind of Internet connection (dedicated, dialup, DSL and ISDN), and boasts very low CPU and memory requirements.
- It has the following features
    - Unlimited domains
    - Unlimited mailboxes
    - Unlimited mailing lists
    - Fast and easy to use interface
    - Runs on Microsoft Windows NT and Microsoft Windows 98
    - Very low CPU and memory requirements
    - Compatible with most mail client software

The 1st Class Mail Server gives all these features for FREE

SmartMail Server *(by VirtualZone.de)*

- It is a reliable and easy-to-use E-Mail-Server which brings the power of e-mail to home or business network.
- Providing all users in a network with e-amil addresses or hosting the mailserver for own website, the full-featured solution SmartMail Server will be the right selection
- SmartMail server capabilities:
    - Serving (all) users on a network (LAN) with e-mail
    - Mail server for Internet-domain

2.4 Overview of Existing Systems

2.4.1 myMail.ro[6]

Web Address: http://www.mymail.ro

WAP Address: http://wap.mymail.ro

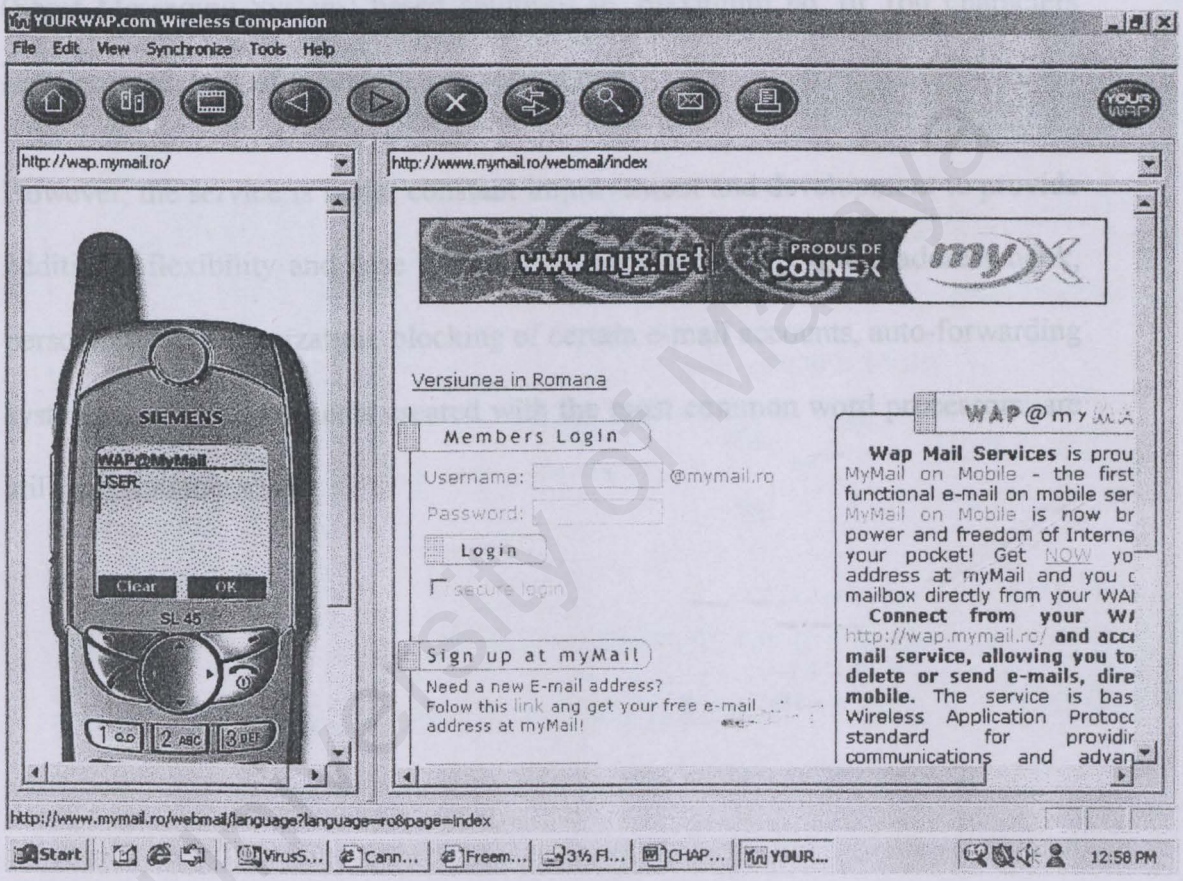


Fig 2-6 myMail.ro E-mail Login Page: WAP & Web Browser View

This service effectively combines the power and mobility offered by the new WAP technology in the mobile telephony industry with traditional e-mail services. The idea is to give access from a WAP mobile handset to personal email

account offering in this way maximum flexibility and mobility. This means that wherever you are you can now use just your mobile phone to read your email, reply to an email or send a new email.

It has all the standard mail functions i.e. read, delete, compose, forward, reply, and the messages can have any size, clearly overcoming the shortages of SMS (Short Messaging System) based solutions ie. maximum no. of 160 characters sent/received, lack of functions, easiness of use.

However, the service is under constant improvement and development to provide additional flexibility and ease of use. Functions such as personal address book, personal folder organization, blocking of certain e-mail accounts, auto-forwarding system, read of attachments created with the most common word processors; are still not available.

2.4.2 YOURWAP.com[5]

Web Address: http://www.yourwap.com

WAP Address: http://wap.yourwap.com

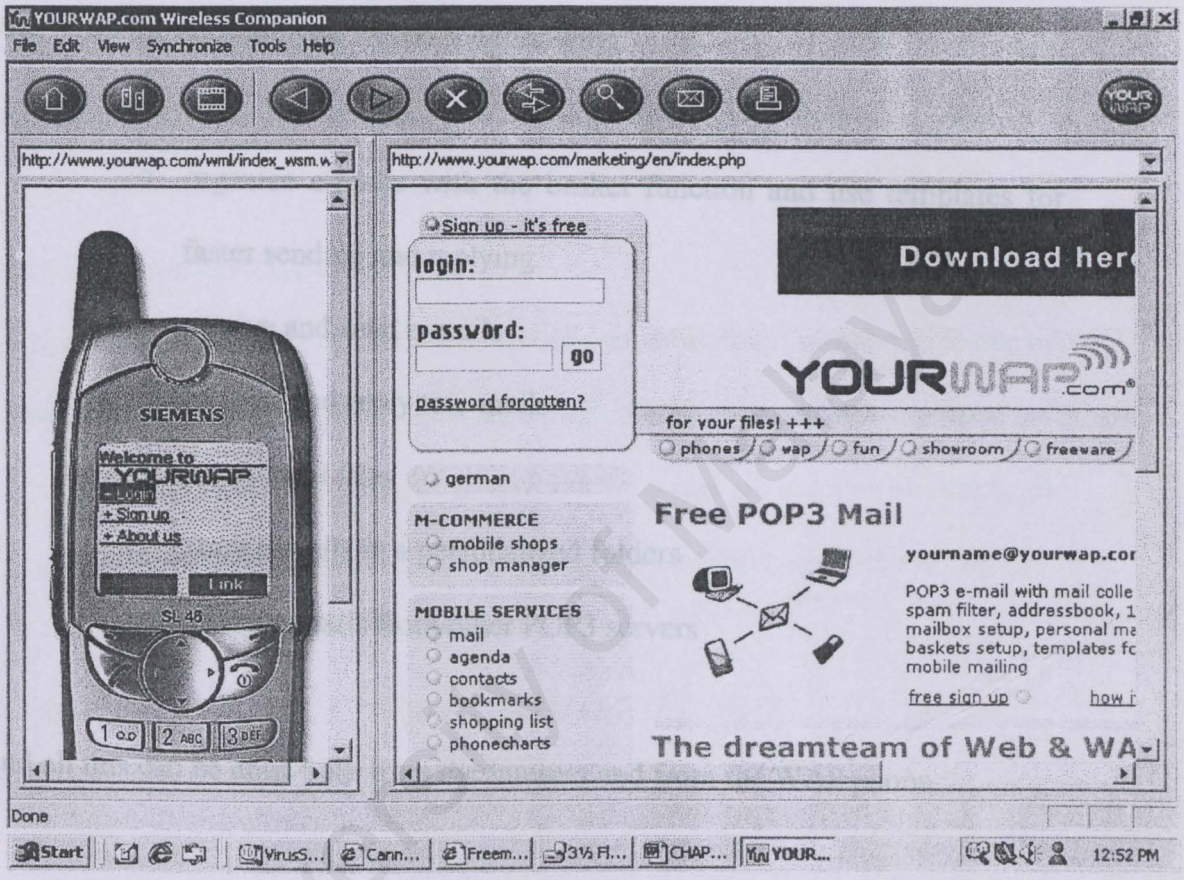


Fig 2-7 YOURWAP.com E-mail Login Page: WAP & Web browser view

The YOURWAP.com mobile and wireless service is powered by o3sis IT AG. By the end of 1999 the YOURWAP.com service was launched. After 1 1/2 years the service now offers a complete suite of Wireless Applications that are already used by people in 124 countries all over the world.

Its mobile services include Mail, Agenda, contacts, Bookmarks, Shopping List, Phonecharts, Notes and Files.

At YOURWAP.com, you will receive a free e-mail account if you sign-up for one. With the account, you can do the following:

- organize e-mails with the basket function and use templates for faster sending and replying
- receive and send e-mails
- forward and reply to e-mails
- attach local files
- store e-mails in a personalized folders
- collect e-mails from other POP3 servers

And all this can be done both from the Internet and from the WAP phone.

2.5 Conclusion

As a whole, literature review is very important to the development of a project as all information relevant to the project are thoroughly analyzed and reviewed. Next, we shall move on to Chapter 3 where we discuss the methodology for the project, as well as recognizing the requirements of the system.

CHAPTER 3: METHODOLOGY

3.1 Concept of Methodology

Methodologies provide comprehensive guidelines to follow for completing every activity in the system development life cycle (SDLC), including specific models, tools, and techniques. A methodology might be homegrown or purchased from a consulting firm.

3.1.1 System Development Life Cycle (SDLC)?

All system development, regardless of specific methodology, is based to lesser or greater degrees on the SDLC. The SDLC describes and defines a process of system development. The process is defined in terms of a series of steps, or phases, that are accomplished over time in order to develop a specific system.

The system development life cycle process has six (6) phases:

Phase I - System Initiation

Phase II - System Analysis

Phase III - System Design

Phase IV - System Building

Phase V - System Implementation

Phase VI - System Maintenance

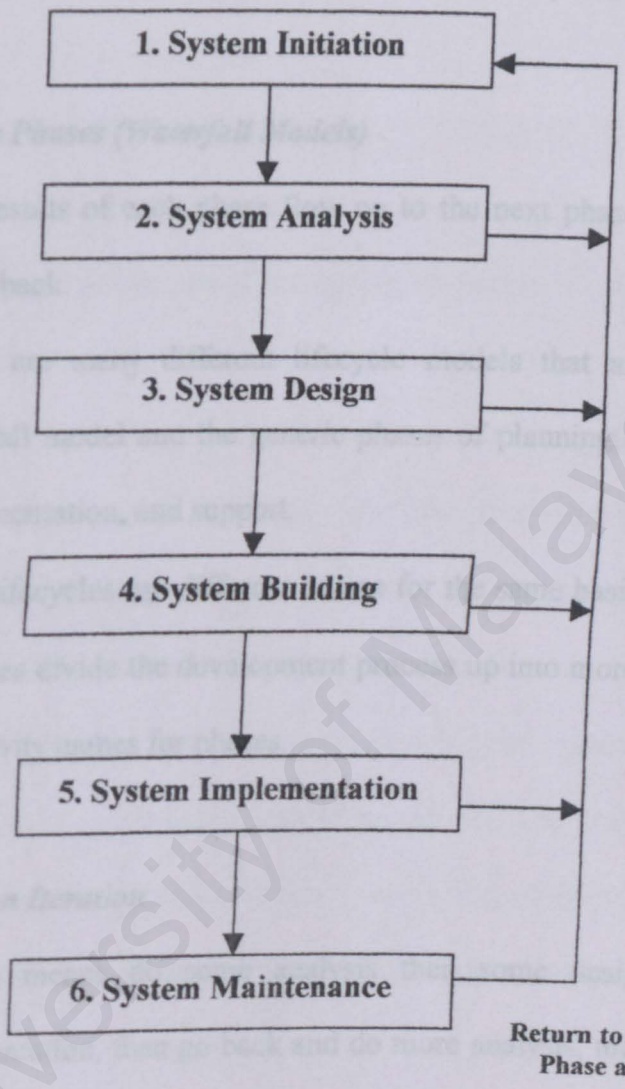


Figure 3.1 Six-Phase System Development Life Cycle (SDLC)

3.1.1.1 System Development Life Cycle Variations

There are some variations to the generic SDLC that are important to understand.

Variations of the Phases (Waterfall Models)

- The results of each phase flow on to the next phase and there is no going back
- There are many different lifecycle models that are based on the waterfall model and the generic phases of planning, analysis, design, implementation, and support.
- Some lifecycles use different names for the same basic phases. Some lifecycles divide the development process up into more phases. Some use activity names for phases.

Variations Based on Iteration

- Iteration means do some analysis then some design then some implementation, then go back and do more analysis, more design, and more implementation.
- Iterative development assumes cannot get it all right the first time through. Some design and some implementation have to be done before going back and complete the analysis.
- This is different than overlapping the phases. Iteration can be effective in improving the quality of the finished system, but there are some

risks. It is harder to manage the project and harder to state exactly where the project is in terms of tasks completed.

Variation Based on an Emphasis on People

- Some methodologies and resulting lifecycles place more emphasis on people and the impact of the system on people, viewing the system as a **sociotechnical system** – information systems that include both social and technical subsystems designed to work well together.
- Other terms used for this include user-centered design and participatory design. Because people are so important to the success of the system, users are heavily involved in the project throughout.
- Multiview and the soft systems methodology are examples. Followers argue that it is relatively easy to model the data and processes in a system, but the people and their conflicting objectives, perceptions, and attitudes make the system's behavior unpredictable.

Variations Based on Speed of Development

- **Rapid application development** (RAD) aims to speed up the system development process using a variety of techniques. Most methodologies include some RAD techniques.
- ***Prototyping*** and ***Joint Application Development*** meetings are techniques commonly used.

3.1.2 Comparisons between *Waterfall* and *Prototyping* Methodologies

The *Waterfall* and the *Prototyping* methodologies were further analyzed in detail to see which is more suitable for the WAPES project. The table below compares the differences between the *Waterfall* and the *Prototyping* methodologies.

Waterfall	Prototyping
1. Forces analysis and planing before actions are taken.	1. Models final product and attributes testing of the final product even if it's not ready yet.
2. ALL details must be defined up front.	2. Prototype models are tested and reviewed throughout development
3. After final requirements are released, there is no room for correction.	3. An iterative process
4. Emphasis on **planning**	4. Emphasis on **users**

Table 3-1: Waterfall vs. Prototyping

3.1.3 Advantages and Disadvantages of *Waterfall* and *Prototyping*

Advantages of *Waterfall* model:

- It forces a discipline process to avoid the pressures of writing code long before it is known what is to be built.
- It is much easier to build something if it is known what that something is.

Disadvantages of *Waterfall* model:

- There is no room for mistakes and no process for error correction after the final requirements are released.
- The waterfall methodology builds products that, by the time they are delivered, are obsolete.
- There is no early feedback from the customer.
- Many times, once the customers see what they could get, the customers want something entirely different then what they said they wanted in the first place.
- The waterfall methodology puts so much emphasis on planning that in a fast moving target arena it can not respond fast enough to change.

Advantages of *Prototyping* model:

- Reduces development time.
- Reduces development costs.
- Requires user involvement.
- Developers receive quantifiable user feedback.
- Facilitates system implementation since users know what to expect.
- Results in higher user satisfaction.
- Exposes developers to potential future system enhancements.

Disadvantages of *Prototyping* model:

- Can lead to insufficient analysis.
- Users expect the performance of the ultimate system to be the same as the prototype.
- Developers can become too attached to their prototypes
- Can cause systems to be left unfinished and/or implemented before they are ready.
- Sometimes leads to incomplete documentation.
- If sophisticated software prototypes (4th GL or CASE Tools) are employed, the time saving benefit of prototyping can be lost.

3.1.4 Methodology for the WAPES Project

Due mainly to the time constraints and the nature of this system, the **prototyping** methodology is being used for this project. A **prototype** is a smaller-scale, representative or working model of a proposed design for an information system. Figure 3.1 is a Data Flow Diagram of the prototyping approach.

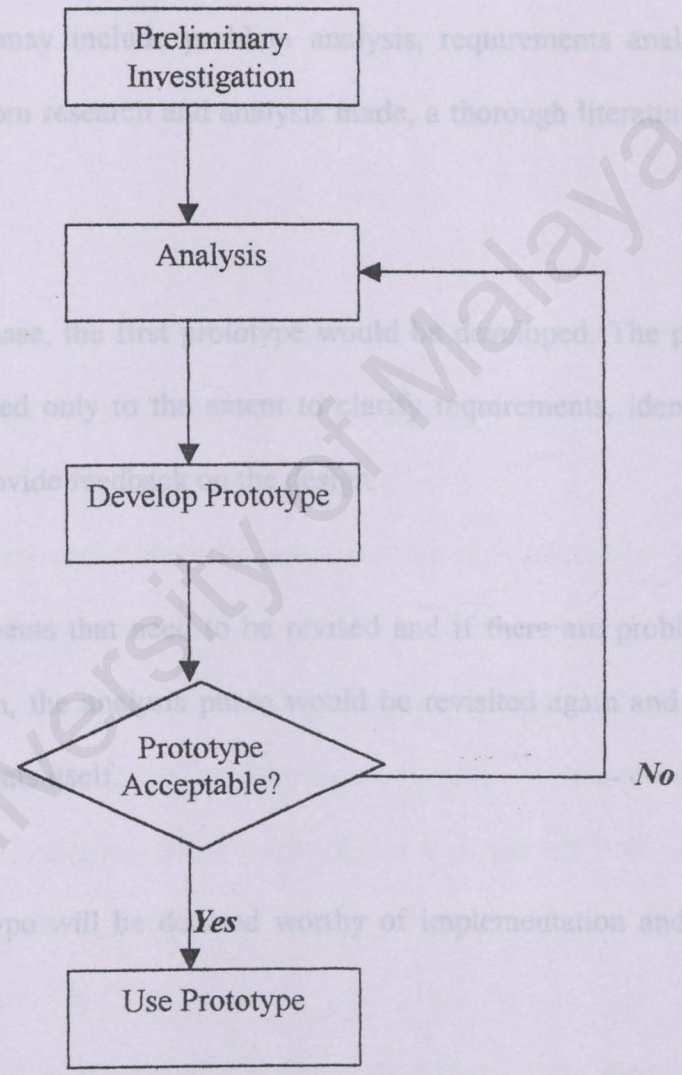


Figure 3.2 Prototyping

Figure 3.2 depicts the development of a prototype. As a start, after the project title is identified, preliminary investigation would be done. The activities include listing problems, opportunities, and directives, defining project objectives and scopes, and planning of the project.

Next, research and analysis would be done to seek all information related to the project title. These may include problem analysis, requirements analysis and decision analysis. From research and analysis made, a thorough literature review would be presented.

After the analysis phase, the first prototype would be developed. The prototype would be implemented only to the extent to clarify requirements, identify new requirements, and provide feedback on the design.

If there are requirements that need to be revised and if there are problems and faults identified, then, the analysis phase would be revisited again and thus the prototyping loop repeats itself.

Eventually, a prototype will be deemed worthy of implementation and thus, is placed into operation.

3.2 Information Sources

There were several approaches undertaken to gather information on WAP. The sources of information can be divided mainly into four categories :

- Internet
- Articles from Media
- Discussion Group
- Books

3.2.1 Internet

Internet has played a major role in succeeding the information retrieval process because it has vast resources and information on the subject.

From the Internet, there are a lot of websites and tutorials related to WAP that could be found using web portals. A **Web portal** (or simply a portal), is a "cyber door" on the Web; it serves as a customizable home base from which users do their searching, navigating, and other Web-based activity. Web portals provide links to many WAP resources. Some examples of web portals that were used include **Catcha, Cari, MSN, Excite and Yahoo!**

Besides web portals, many useful web sites were also found using search engines. A **search engine** is a special kind of Web page software that finds other Web

pages that match a word or phrase users entered. Some examples of search engines being used are **Google, Yahoo!, GoTo and AltaVista.**

3.2.2 Articles from Media

Media such as newspapers and magazines have been a great help in finding information about WAP. Figure 2-1 is a table of the newspaper names, IT column names and respective publication days of the IT columns.

NEWSPAPER	IT COLUMN	DAYS
The Star	*In.Tech*	*Tuesday*
New Straits Times	*Computimes*	*Monday, Thursday*

Both of these newspapers have at least 1 WAP article in each issue, it's latest updates and trend, it's progress locally and internationally.

Besides, these daily newspapers, online newsletters also provide a good source of information and also review on certain technologies. For example, **wapnews** from http://wap.com, **InformIT News** from http://informit.com, and etc.

Magazines also provide a good source of WAP information. Some examples include **PC World, PC.com, netv@lue** (a pullout from **The Edge**) and **Asia Computer Weekly**.

3.2.3 Discussion Group

Online discussion groups provide a great resource for finding answers to WAP questions. With a worldwide network of community members, these discussion groups provide all types of people that can answer questions that both the newbie and experienced WAP developer have. Some discussion groups that I have joined include **WAP Developer Mailing List** from http://anywhereyougo.com and **WAP.com Discussion.**

3.2.4 Books

The main reference book that I have been using is **Professional WAP** from WROX Publication. It provides me with all technical knowledge that I need to know to develop WAP.

3.3 System Requirement

.3.1 Functional Requirements

A functional requirement is a function or feature that must be included in an information system to satisfy the business need and be acceptable to the users. [1]

Below are the functional requirements of the WAPES project:

- The system should enable users to check their e-mails via Internet or WAP browser
- The system should be able to identify login users
- The system should be able to compose, reply, forward and delete e-mails

.3.2 Non-Functional Requirements

A non-functional requirement is a description of the features, characteristics, and attributes of the system as well as any constraints that may limit the boundaries of the proposed solution.

Below are the non-functional requirements of the WAPES project:

- Fast performance – the system should be able to perform efficiently for every users request
- User Interface – the system must be user friendly so that users can use the system comfortably
- Security – the system must be safe and maintains the secrecy and integrity of e-mails

3.4 Tools Selected

3.4.1 WAP Development Toolkit

All emulators have the same basic functions. The only differences lie in their user interfaces.

For this project, **Nokia's WAP Toolkit, Version 2.1,** would be used. This is because it could support WAP Version 1.1 and Version 1.3 as well. Most of the other available emulators only supports Version 1.1.

3.4.2 Development Platform

Windows OS would be used as most of the emulators or development toolkits, can only be run on Windows platform. Furthermore, Windows OS has a more 'user-friendly' GUI (Graphical User Interface) than Unix or Linux.

Therefore, **Windows 2000 Professional** would be used in this project. Windows 2000 Professional was chosen instead of the other Windows OS mainly because it comes with IIS (Internet Information Server).

3.4.3 Web Server

For this project, the WAP Server is not used as this project is just a simulation application. Instead, a Web Server is being used.

Although the Apache Web server is free, and can support more OS, it was not chosen for this project mainly because it is not user-friendly.

As for the Personal Web Server (PWS), it is a scaled-down version of the Internet Information Server (IIS), unable to perform as professionally as the IIS. It is designed for small-scale web-development and therefore is not suitable for this application.

Therefore, **Internet Information Server (IIS)** was chosen instead. Most organizations' systems run on Windows NT platform, that is, easier access and familiarity with IIS. IIS is the professional, commercial Web Server found on many hosting services.

Apache	**PWS**	**IIS**
Free software	*License software*	*License software*
Not user friendly	***User friendly***	***User friendly***
Free open source code	*No free open source code*	*No free open source code*
No GUI	***GUI***	***GUI***

Table 3-2 Comparisons of Web Servers

3.4.4 Dynamic Scripting Language

JSP is a platform independent scripting language if compared to ASP. It supports the concept of source-reuse where its class file can be reused in certain condition without writing the whole codes all over again, that is separating of logic from display.

As such, JSP has been chosen for this project.

JSP	***ASP***
Free software	***License software***
Support many databases	***Support few databases***
Support more OS	***Support Windows Base OS***
Platform Independent	***Platform Dependent***
Sources Reuse	*Not sources reuse*
Hard to learn	*Easier to learn*
Support more OS	***Support windows base OS***
Slow	*Fast*
Easier to manage	*Easy to manage*

Table 3-4 Comparisons of Dynamic Scripting Language

3.4.5 Application Programming Interfaces (APIs)

The JavaMail API offers a clean object-oriented framework of classes that model a theoretical mail system. JavaMail is platform-independent and protocol-independent, and therefore presents an ideal way to build e-mail and messaging solutions that will work with WAP technology.

Many applications can benefit from e-mail support; using JavaMail, developers can rapidly construct messaging functionality whilst abstracting the underlying vendor's implementation in a similar way to the abstraction achieved from relational databases with JDBC.

In many ways the JavaMail service providers act in a similar way to those that provide JDBC drivers, doing for SMTP, Lotus Notes etc, what JDBC does for Oracle and Sybase.

> **JavaMail models an abstract messaging service and any vendor specific mail system that provides a JavaMail interface implementation can be accessed with minimal –(ideally no) – recoding.** [2]

Therefore, JavaMail is being used in the WAPES project. However, by using the JavaMail API, we need to install the following tools:

- JDK (Java Development Kit) 1.2.2. Download from http://java.sun.com/jdk/
- JSDK (Java Servlet Development Kit) 2.1. Download from http://java.sun.com/products/ (This is also available in the JSWDK download)
- JAF (Java Activation Framework). Download from http://java.sun.com/beans/glasgow/jaf.html
- Java Mail 1.1.3, which includes an SMTP service provider. Download from http://java.sun.com/products/javamail
- Java Mail POP3 provider. Separate download from http://java.sun.com/products/javamail/pop3.html

- Access to a SMTP server and a POP3 server is also needed.
- Access to a web server and also a simulator is also needed.

3.4.6 Mail Server

The 1st Class Mail Server was chosen for the WAPES project. This is mainly because it has a very comprehensive and detailed manual and help content as compared to the SmartMail Server. Furthermore, it has more features and capabilities compared to the SmartMail Server.

3.5 Conclusion

In this part of the report, the system methodology has been discussed in detail, system requirements has been identified, as well as tools for the project has been selected. Next, we shall move on to Chapter 4, where we discuss the system design.

CHAPTER 4: SYSTEM DESIGN

4.1 Introduction

System design is a process where system requirements are translated into a representation of software. In other words, system design is defined as the tasks that focus on the specification of a detailed computer-based solution. System design focuses the technical or implementation concerns of the system. [2]

4.2 Process Design

WAPES is designed based on the structured design technique. Structured design is a process oriented for breaking up a large program into hierarchy of structure chart of modules that result in a computer program, easier to implement and maintain. Its synonyms are top-down program design and structured programming. A module is a group of instructions – a paragraph, block, subprogram, or subroutine.

4.2.1 Structure Chart

Structure chart is tree-like diagram. Structure chart modules are depicted by named rectangles. Modules are factored, from top down, into sub-modules. Studying the flow of data through the program derives the structure chart. WAPES is divided into two major components: Web user and WAP user. Each of the two components is further divided into one or many modules. Figure 4-1 and 4-2 show the WAPES structure chart.

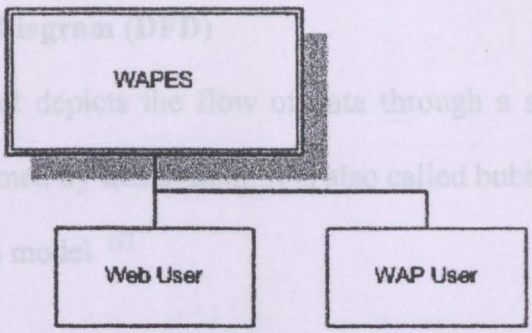


Figure 4-1 Structure Chart for WAPES Main Structure

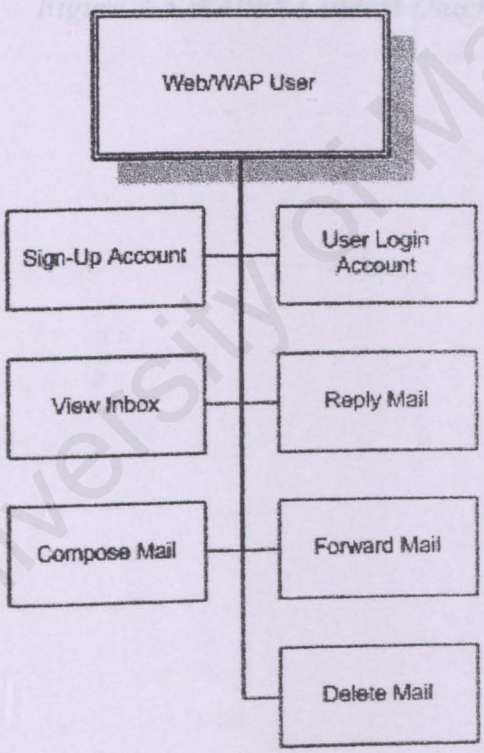


Figure 4-2 Structure Chart for WAPES WEB/WAP User

4.2.2 Data Flow Diagram (DFD)

DFD is a tool that depicts the flow of data through a system and the work or processing performed by that system. It is also called bubble chart, transformation graph and process model. [2]

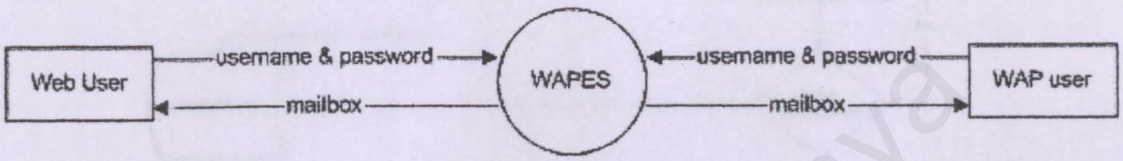


Figure 4-3 WAPES Context Diagram

4.2.3 Flow Chart

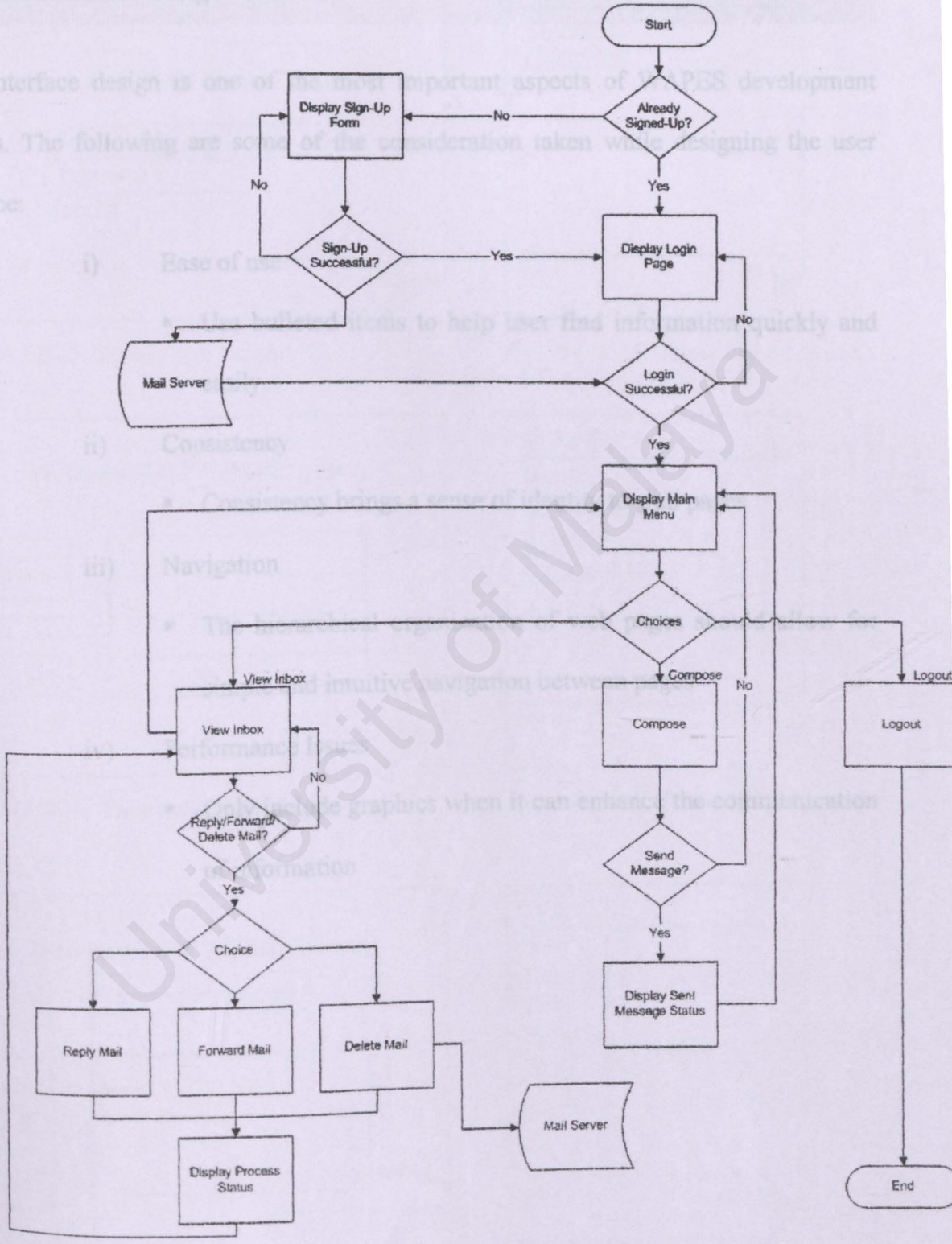


Figure 4-4 Flow Chart for Web and WAP section

4.3 User Interface Design

User interface design is one of the most important aspects of WAPES development process. The following are some of the consideration taken while designing the user interface:

i) Ease of use

- Use bulleted items to help user find information quickly and easily

ii) Consistency

- Consistency brings a sense of identity to web pages

iii) Navigation

- The hierarchical organization of web pages should allow for simple and intuitive navigation between pages

iv) Performance Issues

- Only include graphics when it can enhance the communication of information

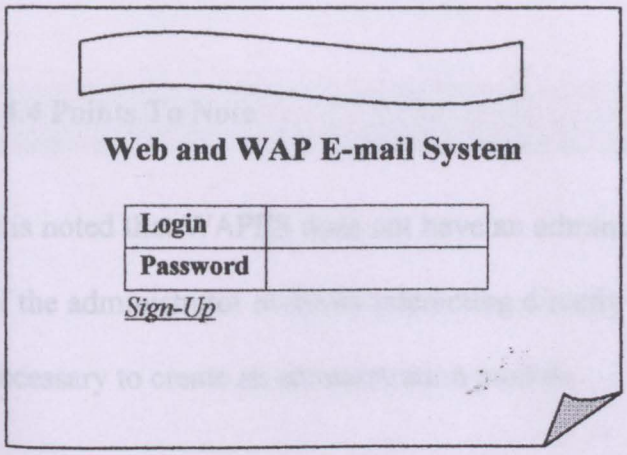


Fig 4-5 Web Home Page Layout

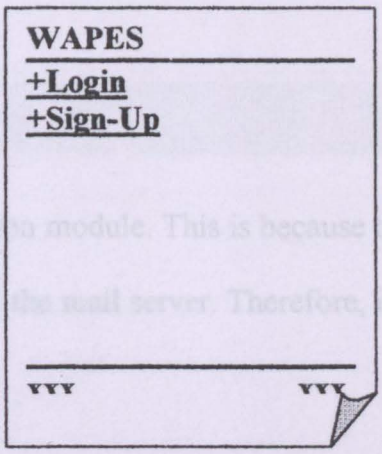


Fig 4-6 WAP Home Page Layout

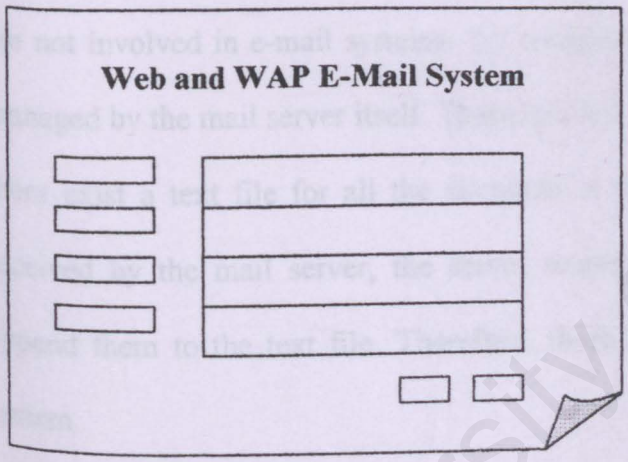


Fig 4-7 Web: View Inbox Layout

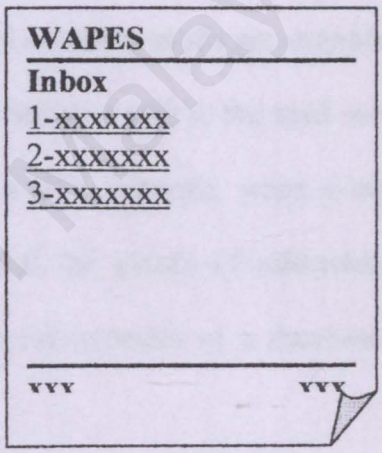


Fig 4-8 WAP: View Inbox Layout

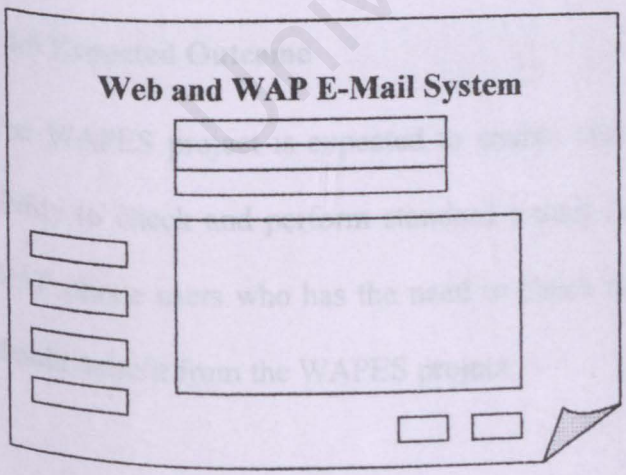


Fig 4-9 Web: Compose/Reply/Forward Layout

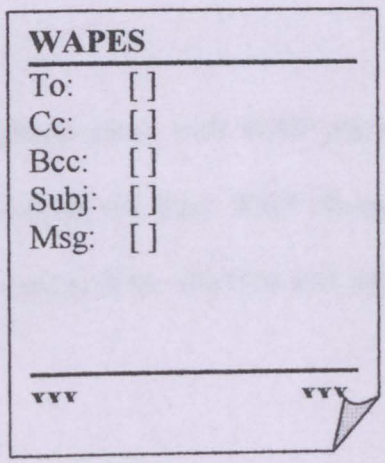


Fig 4-10 WAP: Compose/Reply/Forward Layout

4.4 Points To Note

It is noted that WAPES does not have an administration module. This is because the role of the administrator involves interacting directly with the mail server. Therefore, it is not necessary to create an administration module.

It is also noted that WAPES does not have its database design. This is because databases are not involved in e-mail systems. All incoming and outgoing mails are organized and managed by the mail server itself. There is a list of e-mail accounts in the mail server and there exist a text file for all the accounts in the list. For example, when e-mails are received by the mail server, the server would format the pieces of information and append them to the text file. Therefore, there is no involvement of a database in the system.

4.5 Expected Outcome

The WAPES project is expected to enable mobile phone users with WAP phones, the ability to check and perform standard e-mail functionality via their WAP phones. All WAP phone users who has the need to check their e-mails from anytime and anywhere should benefit from the WAPES project.

4.6 Conclusion

System design is important to be taken into account before any implementation is done, in order to get the overall system flows and to show clearly the ideas on how a system is to be developed.

CHAPTER 5: SYSTEMS IMPLEMENTATION

5.1 Introduction

This chapter describes the processes and strategies of transfroming the system design into workable models and programming codes by setting up the system for actual use (implementation). The implementation of a system involves system coding, testing, installing, documenting, training users, and supporting users.

5.2 System Setup

The system was built using the following hardware and software:

Hardware	Software
Pentium(r) II	Windows 98 Operating System
64.0 MB RAM	Tomcat 3.2.3 Web Server
9.52 GB Hard Disk	Java technology: Java Servlets
NIC (Network Interface Card)	JDK-1.3.1
15" Colour Monitor	JavaMail-1.2
Serial Mouse	Java Activation Framework, JAF-1.0.1
101 Keyboard	Pop3-1.1.1
	Oreilly Package
	Nokia Mobile Internet Toolkit
	Forte for Java CE 3.0
	Macromedia Dreamweaver 3

Table 5.1: System Hardware and Software

5.2.1 Web Server Configuration

1. The 'jakarta-tomcat-3.2.3.zip' file was downloaded from the Internet
2. The zip file was extracted into the `c:/` drive.
3. A folder called 'jakarta-tomcat-3.2.3' with tomcat in it was created
4. To set for Windows9x operating systems:
   a. go to `'c:/autoexec.bat'` file, right-click on the file, choose *Edit*, and type the following in the notepad:

      ```
      set TOMCAT_HOME=c:/jakarta-tomcat-3.2.3
      set JAVA_HOME=c:/jdk1.3.1
      ```

   b. go to the `'c:/config.sys'` file, right-click on the file, choose *Edit*, and type the following in the notepad:

      ```
      shell=\command.com /E:2048/P
      ```

5. To set for Windows 2000 operating systems:
   a. go to *Start->Settings->Control Panel->System*
   b. click on the *Advanced* tab, click on *Environment Variables*
   c. in the lower section titled *System Variables*, click on *New*
   d. enter the following:

      Variable name = **`TOMCAT_HOME`**

      Variable value = **`c:/jakarta-tomcat-3.2.3`**

   e. press 'OK'
   f. again, click on *New*, enter the following:

      Variable name = **`JAVA_HOME`**

      Variable value = **`c:/jdk1.3.1`**

   g. press 'OK'
6. A command prompt (MS-DOS) window was opened, directory is changed to `'c:\jakarta-tomcat-3.2.3\bin'`, then, the command `'tomcat run'` is typed
7. A new *.conf* file called `'tomcat-apache.conf'` was setup in `'c:/jakarta-tomcat-3.2.3/conf'`

8. In order to create another directory in the server, go to 'c:/jakarta-tomcat-3.2.3/conf/server.xml', and the following is added to the context path:

```
<context path="/wapes">
    docbase="<install_dir>:/wapesdev"
    ...
    ...
</context>
```

9. As so, the URL to the WAPES applications would be:

http://localhost:8080/wapes/servlet/

5.2.2 Java Environment Configuration

1. The following products were downloaded from the Internet:
   a. JDK1.3.1
   b. Java Activation Framework JAF-1.0.1
   c. JavaMail-1.3.1
   d. Pop3-1.1.1
2. Then, the paths and classpaths were set
3. For Windows9x operating systems:
   a. go to the 'c:/autoexec.bat' file, right-click on the file, choose *Edit*, and type the following in the notepad:

```
set path=c:\jdk1.3.1\bin;c:\javasoft\jre\1.3.1\bin;
set classpath=c:\jaf-1.0.1\activation.jar;c:\pop3-
1.1.1\pop3.jar;c:\javamail-1.1.3\mail.jar
```

4. For Windows 2000 operating systems:
   b. go to *Start->Settings->Control Panel->System*
   c. click on the *Advanced* tab, click on *Environment Variables*
   d. in the lower section titled *System Variables*, click on *New*
   e. enter the following:

      Variable name = **path**

      Variable value = *refer no.3*

f. press 'OK'

g. again, click on *New*, enter the following:

Variable name = **`classpath`**

Variable value = *refer no.3*

h. press 'OK'

5.2.3 Setting up the Nokia Mobile Internet Toolkit

1. The toolkit was downloaded from the Internet
2. Intallation is done using the Install Shield Wizard

5.2.4 System Structure Settings

The WAPES source codes are stored in a folder: `/wapesdev`. The folder consists of all the servlets, classes, and images that will be accessed by the Client. The folder is divided into 4 subfolders. The servlet classes are kept in the `/WEB-INF/classes` folder. The image files are stored in the `/images` folder. The `/attachment` folder is the folder used to save the attachment files downloaded from the mail server. The `/upload` folder is the folder that is used to store uploaded files that are to be sent as attachments. Figure 5.1 illustrates the system structure.

Figure 5-1: System Structure

5.2.5 Experience Gained

Initially, there were some problems when setting up Tomcat and the Java Environments. Setting up Tomcat requires that the 'TOMCAT_HOME' and 'JAVA_HOME' be set properly, without a semicolon at the end of the declaration line. Or else, tomcat will not function at all as it will be unable to locate its home. Setting up the Java Environments also need proper attention as the classpath must be lead to all the .jar files individually, one-by-one.

5.3 Coding

The WAPES application is basically built using Java Servlets. HTML and WML pages are coded in the servlets itself, as a response when the servlet processes a request from the client.

5.3.1 Coding approach

It is undeniable that good programming skills will determine a more reliable and maintainable system. Here are some coding approach used to practice good programming skills:

-Readability

The source code for every module is written for easy understandablity. Indentation is used to differentiate each clause such as in the control structure (`for` and `while` loop), exception handling clauses (`try` and `catch` clause) and conditional structure (`if` and `else` clause).

-Naming Technique

A good naming technique is used for easier identification by using meaningful names for variables and functions in the program. All the codes that are written in Java follows the Java programming standard with all the class names in title case, for example

`WAPESWeb`, `WAPESWap`, `HTML` and `WML`. Methods and attributes in classes are named in lowercase except for names that are a combination of a few words. The second word and words will use title case as its first letter to differentiate each words such as `forwardCompose`, `saveFile` & etc.

-Modularity

Modularity is an essential need when programming as it reduces complexity and facilitates change results in easier implementation by encouraging parallel development of different parts of a system.

-Documentation

Internal documentation in the source code was used to provide information that identifies the program, describe its data structures, algorithms, and control flow. The header comment block approach is used to provide the above-mentioned information, The documentation style follows the standard that has been specified to implement *javadoc* tool. Codes below show an example of the header comment block of the program:

```
/**
* WAPESWeb.java
* @author Tan Wang Wee
* Purpose To provide e-mail services to clients
* Created on December 22, 2001, 3PM
*/
```

At the beginning of a method, the comment block will be as follows:

```
/** Handles the HTTP <code>POST</code> method.
* @param request servlet request
* @param response servlet response
*/
public void doGetPost(HttpServletRequest request,
                      HttpServletResponse response)
```

5.3.2 Coding tools

The tool used to code the applications' servlets is Forte for Java Community Edition 3.0, which is an IDE used for coding in Java Language. The reason this tool was chosen, is that it provides code readability by highlighting the codes in different colors.

In order to use this tool with JavaMail, all the appropriate .jar files must be mount to the IDE. If this is not done, the IDE will be unable to compile and run the JavaMail codes. To mount the .jar files, go to *File->Mount Filesystem*, and choose the appropriate .jar files.

5.4 JavaMail Implementation

The JavaMail API is a package for reading, composing, and sending e-mails. It is used to create Mail User Agent (MUA) type programs, similar to Eudora or Microsoft Outlook. Users interact with MUA-type programs to read and write e-mails. MUAs rely on Mail Transfer Agent (MTA) to handle the actual delivery.

The core classes that make up the API are `Session`, `Message`, `Address`, `Authenticator`, `Transport`, `Store`, and `Folder`. All these classes are found in the top-level package for the JavaMail API: `javax.mail`, and `javax.mail.internet` packages.

The WAPES application work with the core parts of the JavaMail API. Some coding and explanations are discussed in the next section.

5.5 WAPES Web-Based Implementation

The Web-based interface of WAPES consists of 8 main modules. They are:

1. Login Module
2. View Inbox Module
3. Read Mail and Attachment Module
4. Reply Module
5. Forward Module
6. Delete Module
7. Send Mail and Attachment Module
8. Logout Module

It is noted that the Sign-Up Module, mentioned in Chapter 4 is not included in the systems implementation. This is because JavaMail has no facilities for adding new user accounts, removing user accounts, and changing passwords for users on the mail server. There are no standards in this area as every mail server handles these functions individually and differently. In addition, the WAPES project is an application that provides users the facility to connect to any POP3 server to retrieve mail. Therefore, the sign-up module is deemed irrelevant.

5.5.1. Login Module

The login module basically creates a mail session. It is through this session that everything else in the application works. The login module gets information like mail server, username and password that can be shared across the entire application. The main engine of the login module is shown below:

```
…
…
//Get the system properties
Properties props = System.getProperties();
props.put("mail.smtp.host", smtphost);
//Get SMTP session
Session smtpSession = Session.getInstance(props, null);
smtpSession.setDebug(false);
//Get POP3 Session
Session pop3Session = Session.getInstance(System.getProperties(), null);
pop3Session.setDebug(false);
//Get POP3 Store
Store pop3Store = pop3Session.getStore("pop3");
pop3Store.connect(pop3host, username, password);
…
…
UserSessionData usd = new UserSessionData(smtpSession, pop3Session, pop3Store,
username);
…
…
```

Figure 5-2: 'Login' Code

5.5.2. View Inbox Module

The view inbox module connects to the POP3 Store, to get the only available folder, that is the INBOX. For POP3, the only folder available is the INBOX, unlike IMAP, where there are other folders available.

```
//Get Folder
Folder folder = userSessionData.getPop3Store().getFolder("INBOX");
folder.open(Folder.READ_ONLY);
//Get Directory
Message message[] = folder.getMessages();
int n = message.length;
int totalMessages = folder.getMessageCount();
...
//Listing out the messages
for (int i=n-1; i >= 0; i--) {
                String emailAddress =
((InternetAddress)message[i].getFrom()[0]).getAddress();
                html.println("<tr>" +
               "<td width=\"35%\"><nobr>" +
                "<div align=\"left\"><font size=\"2\">" + emailAddress +
"</font></div>" +
                "</nobr></td>" +
                "<td width=\"55%\"><nobr>" +
                "<div align=\"left\"><font size=\"2\">" +
                "<a href=\"" + request.getRequestURI() + "?action=read&index="
+ (i + 1) + "\">" +
                message[i].getSubject() +
                "</a><br>" +
                "</font></div>" +
                "</nobr></td>" +
                "<td width=\"10%\"><nobr>" +
                "<div align=\"center\"><font size=\"2\">" +
                "<a href=\"" + request.getRequestURI() +
"?action=delete&index=" + (i + 1) + "\">" +
                "Delete" +
                "</a><br>" +
                "</font></div>" +
                "</nobr></td>" +
                "</tr>")
      }
          ...
      ...
```

Figure 5-3: 'View Inbox' Code

5.5.3. Read Mail and Attachment Module

Reading mails consists of reading messages with message contents in text (string), multipart (attachments), or input stream.

There were not much problem coding for reading text messages, which are of MIME type 'text/plain' or 'text/html'. Basically, the module would display the string in its preformatted form in the browser.

However, initially, there were problems reading multipart messages which are of type 'multipart/* '. For each multipart messages, it consists of 2 or more bodyparts. For each bodyparts, the module would check for its content type, and if the content type is an instance of string (text), then, it would display the string in the browser. But, if the content type is not an instance of string, then, that is where the problem arises. The solution to this problem was to save the file in a temporary folder in the server, and to create a link in the browser for the user to retrieve the files, enabling the application to handle attachment files properly. Figure 5-4 below illustrates the saveFile() method used:

```
public String saveFile(String filename,
    InputStream input) throws IOException {
        if (filename == null) {
            filename = File.createTempFile("att", ".tmp").getName();
        }
        // Do not overwrite existing file
        File file = new File("../wapes/attachment/" + filename);
        for (int i=0; file.exists(); i++) {
            file = new File(filename+i);
        }
        FileOutputStream fos = new FileOutputStream(file);
        BufferedOutputStream bos = new BufferedOutputStream(fos);
        BufferedInputStream bis = new BufferedInputStream(input);
        int aByte;
        while ((aByte = bis.read()) != -1) {
            bos.write(aByte);
        }
        bos.flush();
        bos.close();
        bis.close();
        return filename;
    }
```

Figure 5-4: saveFile() method

Lastly, for message content of type input stream, it is displayed using the output stream method of the servlet.

Here are the codes to sort a message according to content-type, and ways of handling each of the types:

```
…
if(messageContent instanceof String){
      output.println("<pre>" + (String)messageContent + "</pre>");
}else if (messageContent instanceof Multipart){
      Multipart mp = (Multipart)messageContent;
            int count = mp.getCount();
            for (int i=0; i<count; i++)
            {    Part part = (mp.getBodyPart(i));
                 String disposition = part.getDisposition();
            if ((disposition != null) &&
            ((disposition.equalsIgnoreCase(Part.ATTACHMENT))||(disposition.equ
            alsIgnoreCase(Part.INLINE)))) {
                                 String filename = part.getFileName();
                           String newfilename = saveFile(filename,
part.getInputStream());
                                 …
                            }else if (disposition == null){
                                 MimeBodyPart mbp = (MimeBodyPart)part;
                                 Object partContent = mbp.getContent();

                                  if (partContent instanceof String){
                                                output.println("<pre>" +
(String)partContent + "</pre>");
                                                }else{
                                               String filename = part.getFileName();
                                               String newfilename = saveFile(filename,
            part.getInputStream());
                                 …
                                               }
                  }
               }
}else if (messageContent instanceof InputStream) {
      OutputStream out = response.getOutputStream();
            InputStream is = (InputStream)messageContent;
            int c;
            while ((c = is.read()) != -1){
                  out.write(c);
             }
}
…
```

Figure 5-5: Read Module Code

5.5.4. Reply Module

Basically, the reply module takes the 'from' parameter of the message and pass it as 'to', and adding a 'Re:' in front of the message subject, and passing it back to the compose method for users to type their reply message.

5.5.5. Forward Module

Forwarding messages is more involved. The message to forward is build up of parts that make up a message. To forward a message, one part is created for the text of the message, a second part is created with the message to forward (by copying over its DataHandler), and combining the two into a multipart, as shown in the codes below.

```
...
//Get Folder
Folder folder = userSessionData.getPop3Store().getFolder("INBOX");
folder.open(Folder.READ_ONLY);
//Get Directory
int messageIndex = Integer.parseInt(request.getParameter("index"));
Message message = folder.getMessage(messageIndex);
DataHandler data = message.getDataHandler();
//Create message to forward
Message forward = new MimeMessage(userSessionData.getSmtpSession());
...
//Create new message part
MimeBodyPart textPart = new MimeBodyPart();
textPart.setText(text);
//Create and fill part for the forwarded content
MimeBodyPart attachForwardPart = new MimeBodyPart();
attachForwardPart.setDataHandler(data);
//Create a multipart to combine the parts
Multipart multipart = new MimeMultipart();
multipart.addBodyPart(textPart);
multipart.addBodyPart(attachForwardPart);
//Associate multi-part with message
forward.setContent(multipart);
//Send message
Transport.send(forward);
folder.close(false);
...
```

Figure 5-6: Forward Module Code

5.5.6. Delete Module

Deleting messages involves working with the Flags associated with the messages. To delete messages, the messages' DELETED' flag is set, as shown in the below codes.

```
...
//Get Folder
Folder folder = userSessionData.getPop3Store().getFolder("INBOX");
folder.open(Folder.READ_WRITE);
//Get Directory
int messageIndex = Integer.parseInt(request.getParameter("index"));
Message message = folder.getMessage(messageIndex);
message.setFlag(Flags.Flag.DELETED, true);
//close connection
folder.close(true);
...
```

Figure 5-7: Delete Module Code

5.5.7. Send Mail and Attachment Module

To support file upload (attachment), form-based upload is used in the HTML form, where the enctype is multipart/form-data. When the 'Send' button is clicked, the browser locates the file in the local system and sends it using HTTP POST , encoded using the MIME-type multipart/form-data. When it reaches the servlet, the servlet must process the POST data in order to extract the encoded file.

Unfortunately, there are no methods in the servlet to do so. The solution was to use a ready-built library to facilitate uploading between the browser and the servlet. The library used was the class MultipartRequest of the O'Reilly package. This package is available free for download. at `http://www.servlets.com/cos/index.html` . The cos.zip file was downloaded and the `cos.jar` file was added to the classpath settings, to be able to use this package.

Important codes to send message is shown below:

```
…
MultipartRequest multi = new MultipartRequest(request, "d:/wapesdev/upload",
1024 * 1024);
Enumeration params = multi.getParameterNames();
String name[] = new String[6];
String value[] = new String[6];
int i=0;
while (params.hasMoreElements()) {
      name[i] = (String)params.nextElement();
            value[i] = multi.getParameter(name[i]);
            i++;
            }
Enumeration files = multi.getFileNames();
String nama = (String)files.nextElement();
String filename = multi.getFilesystemName(nama);
String type = multi.getContentType(nama);
File f = multi.getFile(nama);

//Define message
MimeMessage message = new MimeMessage(session);
…
if (filename != null){
      //Create new message part
            MimeBodyPart textPart = new MimeBodyPart();
            textPart.setText(text);

      //Create and fill part for the forwarded content
            MimeBodyPart attachPart = new MimeBodyPart();
            DataSource source = new FileDataSource("d:/wapesdev/upload/" +
filename);
            attachPart.setDataHandler(new DataHandler(source));
            attachPart.setFileName(filename);

            //Create a multipart to combine the parts
            Multipart multipart = new MimeMultipart();
            multipart.addBodyPart(textPart);
            multipart.addBodyPart(attachPart);

            //Associate multi-part with message
            message.setContent(multipart);
        }
…
…
```

Figure 5-8: 'Send' Code

5.5.8. Logout Module

To logout from the system, the user session data kept in the UserSessionData class is destroyed, as shown below.

```
…
this.getUserSessionData().destroy();
this._userSessionData = null;
…
```

Figure 5-9: 'Logout' Code

5.6 WAPES Wap-Based Implementation

The Wap-based interface of WAPES also consists of 8 main modules. They are:

1. Login Module
2. View Inbox Module
3. Read Mail Module
4. Reply Module
5. Forward Module
6. Delete Module
7. Send Mail Module
8. Logout Module

Basically, all the modules are the same as the modules for WAPES Web-based respectively, except for the 'View Inbox Module', 'Read Mail Module' and the 'Send Mail Module', which would be discussed as follows.

5.6.1. View Inbox Module

As the wap browser is a small device that has limited memory and display capabilities, problem arises when the messages in the INBOX are many. That is, the browser is unable to display all of the messages and prompts an error message as the size is too big. To overcome this problem, if the number of messages in the INBOX are many, only the latest eight messages will be displayed by the browser, as shown by the codes below.

```
…
//Get Folder
Folder folder = userSessionData.getPop3Store().getFolder("INBOX");
folder.open(Folder.READ_ONLY);
//Get Directory
Message message[] = folder.getMessages();
int n = message.length;
if ( n <= 10 ){
      for (int i=n-1; i >= 0; i--) {
            …}
}else if ( n > 10 ){
                  wml.println("<small><i>Only Latest 10 Messages
Displayed!</i></small>");
                  for (int i=n-1; i >= n-10; i--) {
                                    …}
}
…
```

Figure 5-10: View Inbox Code (Wap-based)

5.6.2. Read Mail Module

WAPES Wap-based is unable to read mail size that are more than 1500 bytes, due to the natural display limitations of the wap device. In addition, it is also unable to read attachments. The reasons are as follows:

a. attachment size are usually large
b. wap browser can only handle content type that is an instance of string and not any other types of content type

Therefore, the application will only list the attachment file name to show the existence of the attachments. However, the attachments cannot be displayed.

Besides, message size limitation, implementing the read module involves implementing a method to handle message of content-type 'text/html'. As by default the Wap browser is only able to display 'plaintext' messages, a `Remover()` method was created to sort out the HTML tags, and to return the output as a string. Figure 5-11 illustrates the `Remover()` method.

```
public String Remover (String original) {
        String output = null;
        StringBuffer buf = new StringBuffer();
        System.out.println("original: " + original);
        StringTokenizer st = new StringTokenizer(original, "<");
        // 2 String arguments, the second String is the delimiter String
        // if it doesn't start with a '<', put the first string to the buf
        if(original.charAt(0)!='<')
            buf.append(st.nextToken()+" ");
        while (st.hasMoreTokens()) {
            String tempString = st.nextToken();
            System.out.println("tempString:"+tempString);
            StringTokenizer st2 = new StringTokenizer(tempString,">");
            String tag = st2.nextToken();
            System.out.println("tag:"+tag);
            while (st2.hasMoreTokens()) {
                String text = st2.nextToken();
                buf.append(text+" "); //add a space after each text piece
                System.out.println("text:"+text);
            }
        }
        output = buf.toString();
        System.out.println("output = " + output);
        return output;
    }
```

Figure 5-11: Remover() method

5.6.3. Send Mail Module

There are no functions implemented for sending attachments from a wap device. This module is implemented such as only text messages are sent.

5.7 Conclusion

The actual implementation of a system requires careful planning to ensure a smooth switch-over. Firstly, installing and configuring software is the basics as without all the basic tools or software working, it is impossible for the implementation of the system to run smoothly. For more information about implementing field and input validation, kindly refer the next chapter: *Chapter 6: System Testing.*

CHAPTER 6: SYSTEMS TESTING

6.1 Introduction

Testing is a verification and validation process. A successful testing will uncover errors in the software and demonstrates that system functions appear to be working according to specification[19] . Glen Myers (1979) states a number of rules that can serve well as testing objectives:

- Testing is a process of executing a program with the intent of finding an error
- A good test case is one that has a high probability of finding an as-yet-undiscovered error
- A successful test is one that uncovers an as-yet-undiscovered error

6.1.1 Testing principles

There is a set of testing principles that should be understand to guide the system testing. Several testing principles suggested by Davis (1995) have been followed in testing the application including:

- All tests should be traceable to user requirements.
- Test should be planned long before testing began. Testing planning can begin as soon as the requirement model is complete.
- Testing should begin 'in the small' and progress toward testing 'in the large'. The first test planned and executed generally focus on individual components. As testing progress, focus shifts in an attempt to find errors in integrated clusters of components and ultimately in the entire system.

6.1.2 Testing Strategies

There are various testing strategies available to assess completeness and correctness of a system. The newly developed WAPES system is tested thoroughly using different testing strategies that involves unit testing and integration testing

As each function module of the system was developed incrementally, the testing of the system was done in two iterative methods. They are unit testing and integration testing.

6.2 Unit Testing

Unit testing focuses on verification effort at the smallest unit of system design - subroutines or functions. Each component is treated as a stand-alone entity and is tested individually to ensure that they operate correctly.

Unit testing involves, examining the code by reading through it, trying to spot algorithm, data, and syntax faults, comparing the codes with the specifications and design, compile the code and eliminate remaining syntax faults, and developing test cases to show the input is properly converted to the desired output[20] .

Unit testing for the WAPES application involved the following:

- ***Reviewing/Examining the code***

  The most basic testing technique is reviewing the code. The important function in each module was commented so that it can be easily traced in the future. Each function in the module was tested on its correctness of logic, systems flow and code syntax.

- ***Using test cases for testing***

  Dummy data were used to test the stability and efficiency of each function unit. This method mirrors real life attributes and will test how far the module can process different types of input.

- ***Testing the boundary conditions***

  Boundary conditions are tested to ensure that the module operates properly at boundaries established to limit or restrict processing

- ***Testing module interface***

  Module interface is tested to ensure that information properly flows into and out of the program unit under test.

- ***Testing each independent paths***

  All the independent paths in a control structure were tested at least once.

- ***Testing all error handling paths***

  All the error-handling paths were being tested to make sure that all the errors were being handled and would not cost any inconvenience to the system.

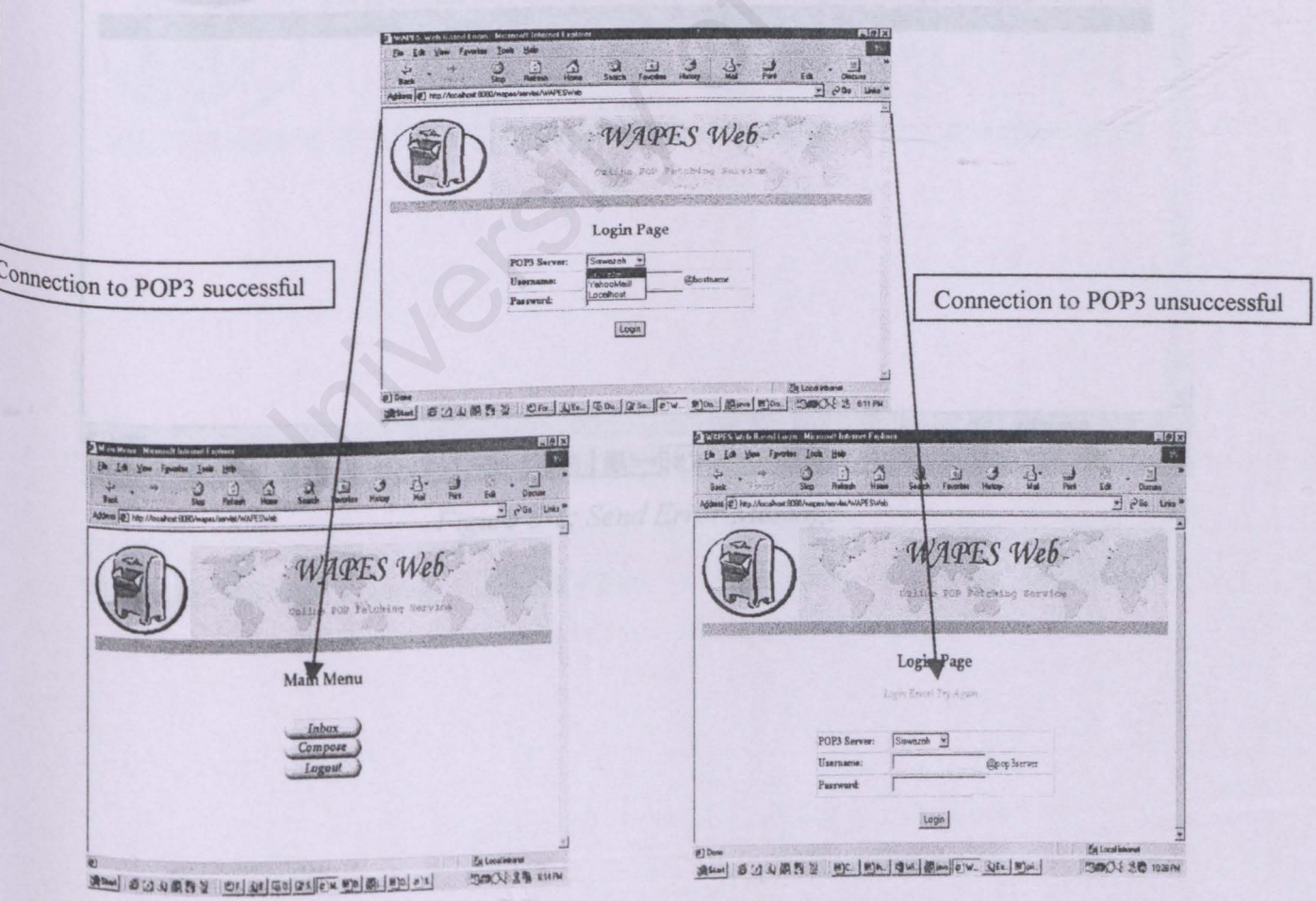
This process enable the detection of errors in coding and ensures that logical mistakes were contained within the boundary of that module. The relative complexity of tests and uncovered errors was limited by the constrained scope established for unit testing.

6.2.1 Testing Server Connectivity

Testing for server connectivity includes connection to web server, POP3 server and SMTP server.

To connect to the web server, the URL : http://localhost:8080 is typed. If the connection is active, the default Tomcat homepage will be displayed. However, if an error message is found, it means that the server would have to be restarted, or if more seriously, to be re-installed.

To connect to a POP3 server, a server is chosen from the drop-down-list and the respective Username and Password to the server is typed. If a connection to the POP3 server is established, the Main Menu will be displayed. However, if connection to the POP3 server is not successful, an error message will be displayed as follows:



Connection to the default SMTP server for WAPES is to the FCSIT's mail server, siswazah.fsktm.um.edu.my. To test whether connection to the server is active, a mail is sent using the Compose Module. If connection to server is active, the status page to show that the send operation is successful will be shown.

However, if there are problems with the connection to the server, an error message will be shown. As an example:

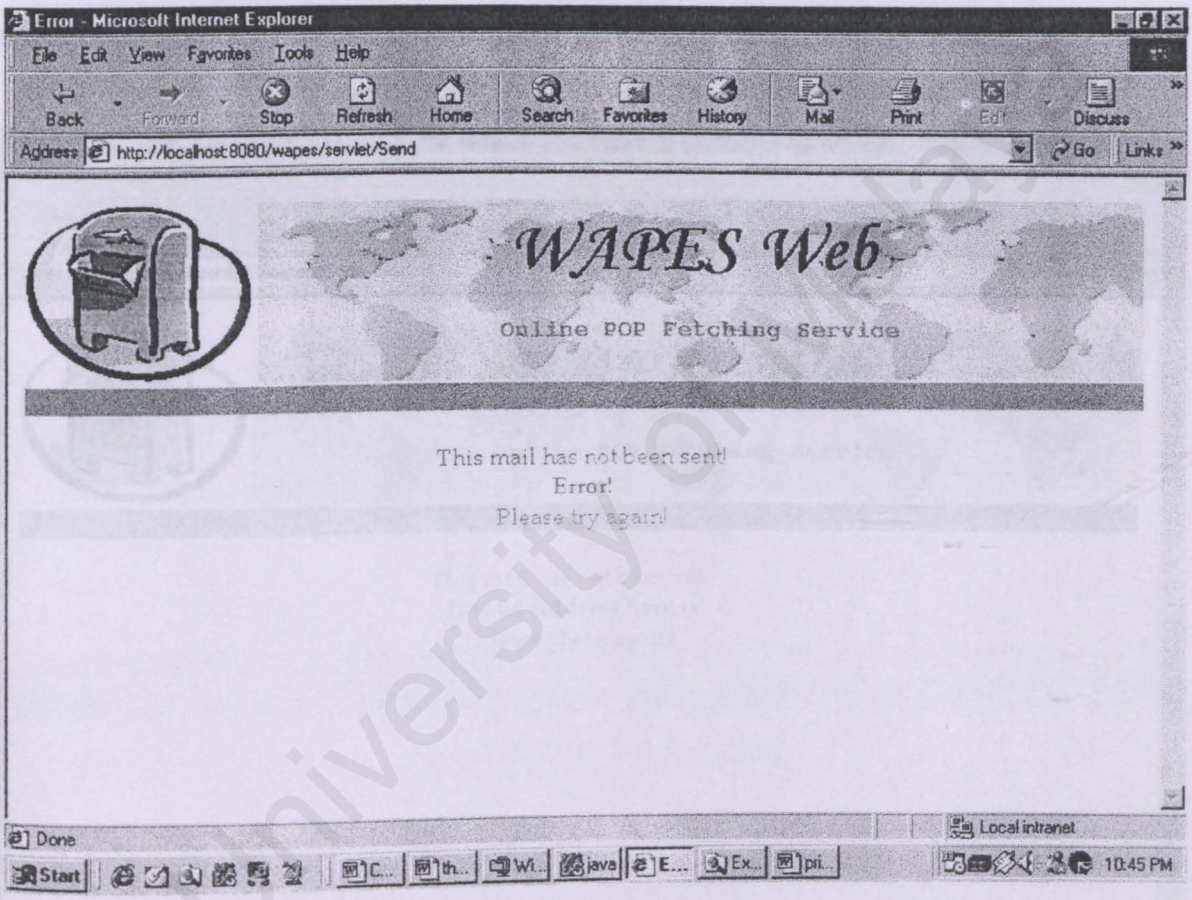


Figure 6-1: Send Error Message

6.2.2 Testing Field Validation

Field validation can be prominently tested in the Compose, Reply and Forward Module. Every address syntax in the input fields must be correct in order that the mail can be successfully sent. This is done using the 'AddressException' exception of the JavaMail API. However, this exception will only test whether the address syntax has some string before and after the '@' symbol. It will not check the existence of the addresses. An error message will be displayed to notify the users to check their address inputs again if there are address syntax errors. As an example, figure 6-2 illustrates the error page.

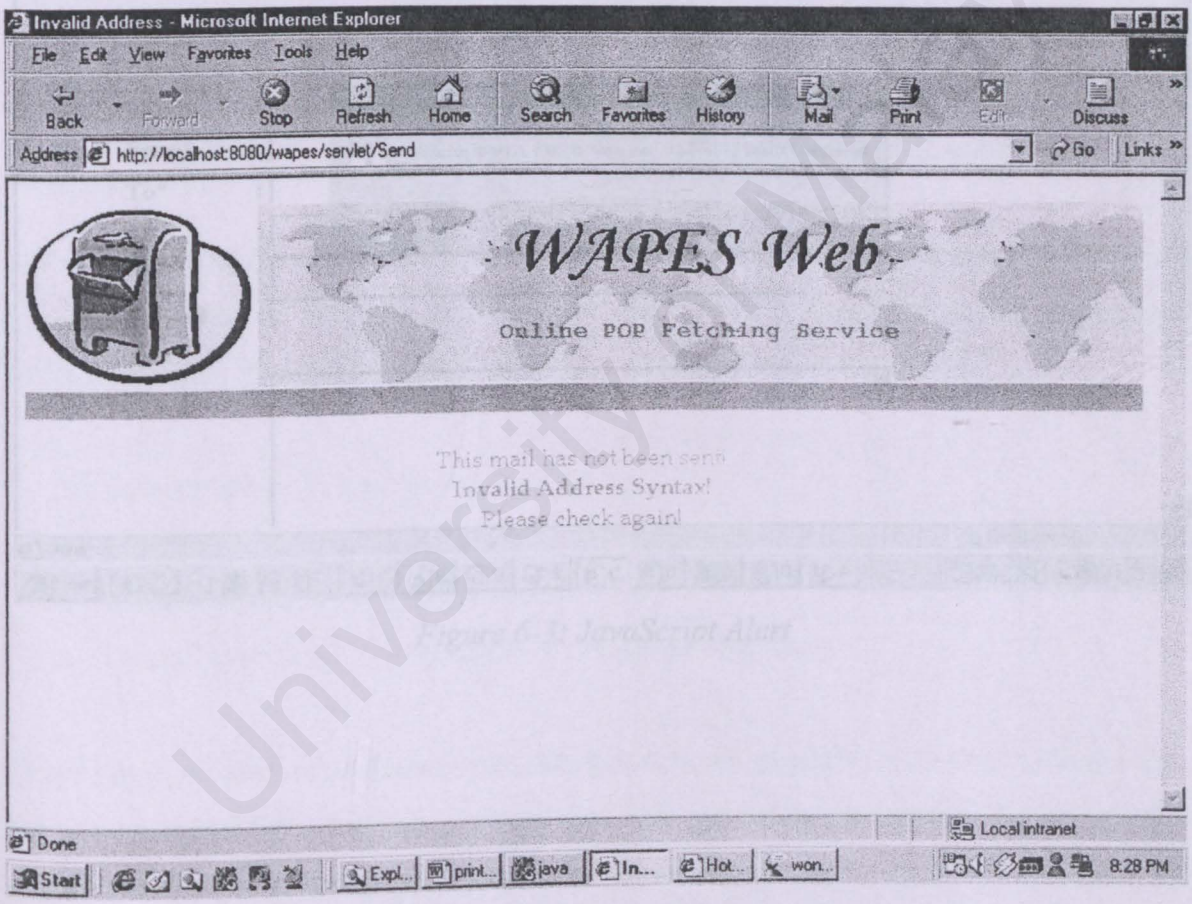


Figure 6-2: Invalid Address Syntax Error Message

Besides address syntax validation, the 'From' and 'To' input fields has some underlying JavaScripts to make sure that these 2 fields are filled up. If either one of these 2 fields are not filled up, an alert will be prompt to remind users. As an example:

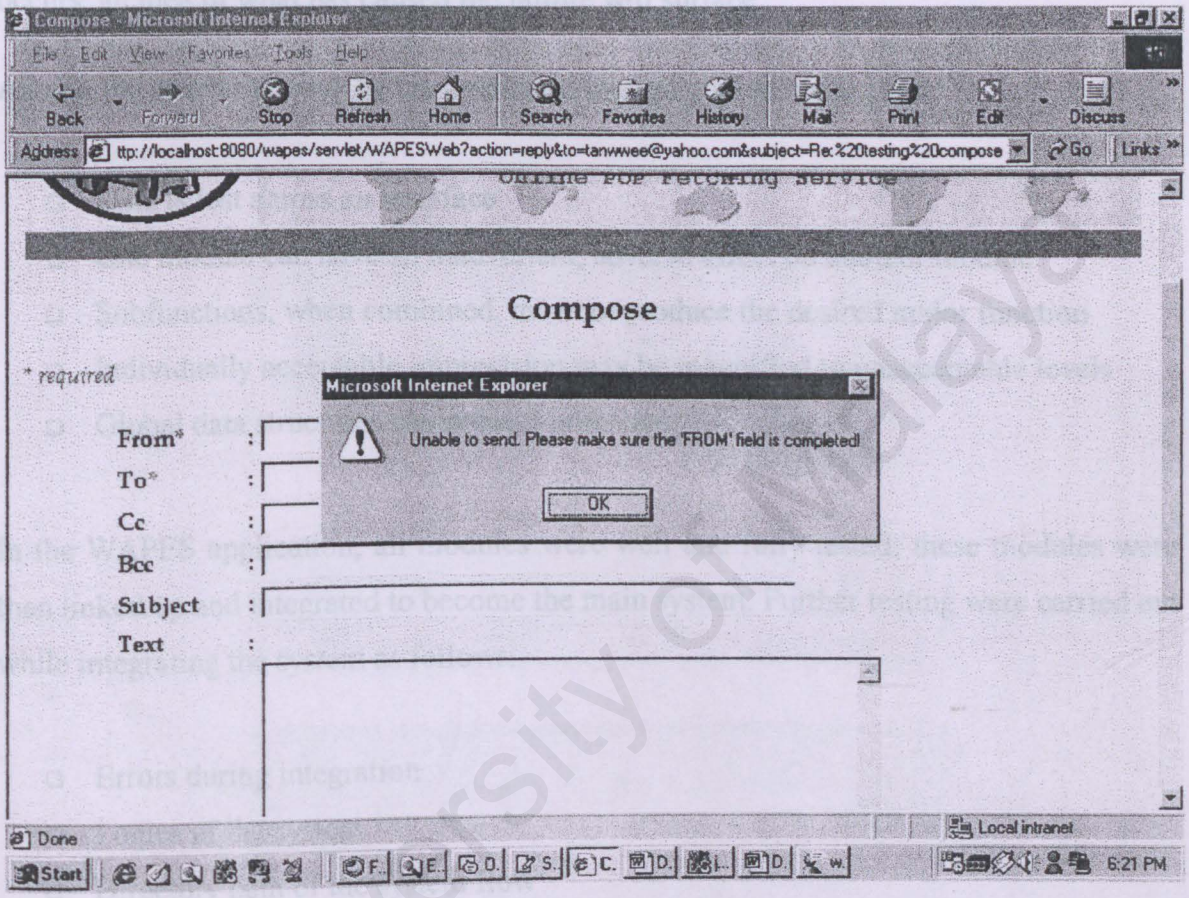


Figure 6-3: JavaScript Alert

6.3 Integration Testing

Integration testing was carried out after individual components were working correctly and have meet objectives. It is done by combining the individual components in a working system. This integration is planned and coordinated so that when a failure occurs, an idea of what has caused the failure will surface.

Among the errors that may be uncovered during integration testing are: [19]

- Data is lost across an interface
- One module can have an inadvertent, adverse effect on another module
- Subfunctions, when combined, may not produce the desired major function
- Individually acceptable imprecision may be magnified to unacceptable levels
- Global data structures can present problems

In the WAPES application, all modules were well and fully tested; these modules were then linked up and integrated to become the main system. Further testing were carried out while integrating the system as follows:

- Errors during integration
- Logics of the system
- Directory path of the system flow
- Overall system whether it meets the required objectives

There are a number of different integration strategies available: top-down integration, bottom-up integration, big-bang integration and sandwich integration. Based on the system characteristics and project schedule, the sandwich testing approach was selected as WAPES' integration testing. This strategy combines the top-down strategy with the bottom-up strategy. The modules were tested individually and then combined together to become the main system and then tested again.

6.3.1 Value Passing

Integration between modules can be prominently tested in the value passing between the 'View Inbox' and 'Read' Module, between the 'View Inbox' and 'Delete' module, and between 'Read' module and the 3 operative modules – 'Reply', 'Forward' and 'Delete'.

When the links in these modules' interfaces are clicked, it is very important that the correct message is displayed, replied, forwarded and deleted. Therefore, the value passing of the 'message index' is very important.

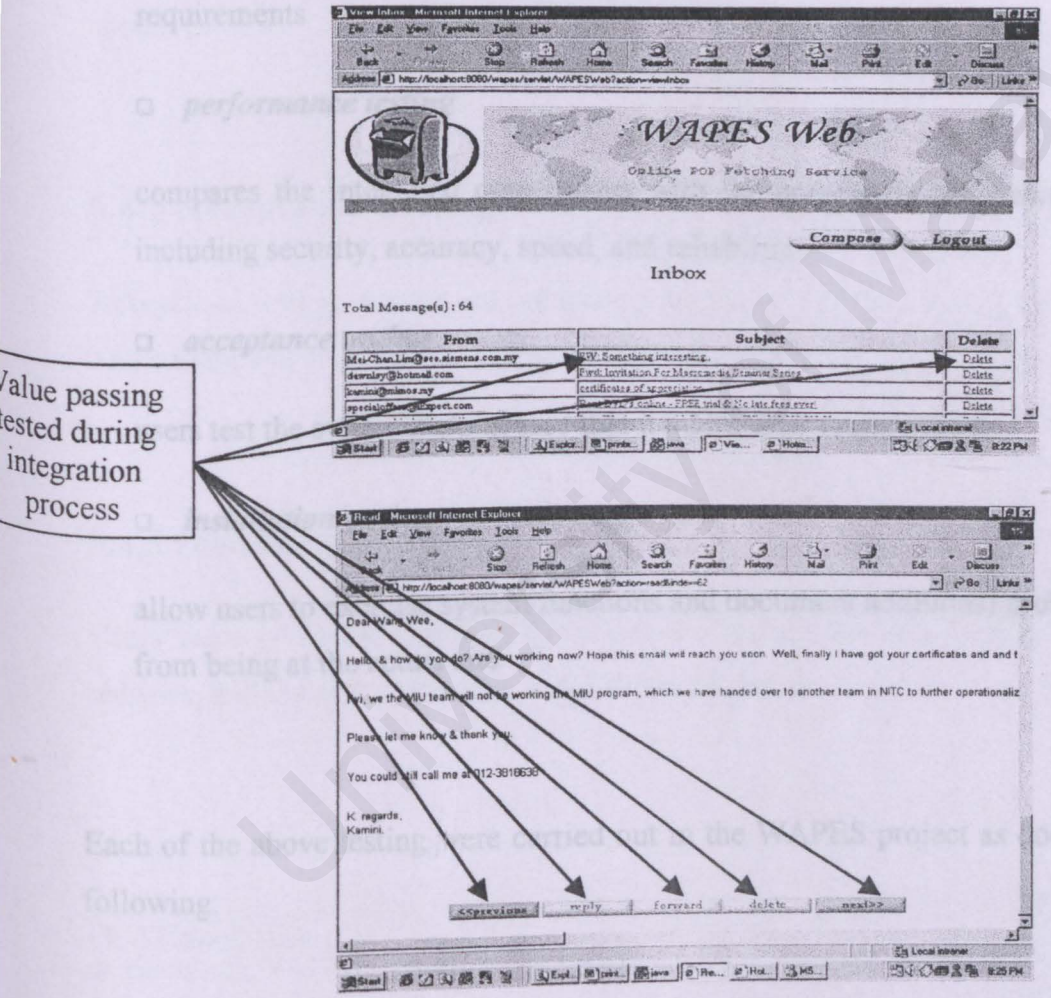


Figure 6-4: Testing Value Passing

These links have been well tested to ensure they link to correct messages and performs the right operation on the right messages.

6.4 System Testing

The objective of unit and integration testing was to ensure that the code implemented the design properly. However, in system testing, the objective is different. System testing ensures that the system does what the user wants to do[20] . System testing process includes the following steps:

- ***function testing***

  checks that the integrated system performs its functions as specified in the requirements

- ***performance testing***

  compares the integrated components with the non-functional system requirements, including security, accuracy, speed, and reliability

- ***acceptance testing***

  users test the system making sure that it meets their requirements

- ***installation testing***

  allow users to exercise system functions and document additional problems that result from being at the actual site

Each of the above testing were carried out in the WAPES project as documented in the following:

6.4.1 Function Testing

The function of the WAPES system is to send and read e-mails. These two main functions have been fully tested for web-based and also wap-based. In order to test that

WAPES is able to provide standard e-mail functionality using mobile phones, test is done using the Nokia Mobile Internet Toolkit Simulator, and also the YOURWAP Wireless Companion Simulator. It has already been proven that WAPES is able to perform its functionality as stated in the objective.

6.4.2 Performance Testing

The system was found to have a friendly interface, and has an average performance for its speed.

6.4.3 Acceptance Testing

Acceptance testing was carried out where the system was tested out by a group of friends. The system was found to be suitable to their (users) needs.

6.4.4 Installation Testing

Users were given the opportunity to test the system not only locally but through the network, and the system manage to fulfill the requests.

6.5 Conclusion

Testing of computer systems is a major task as it validates the requirements to ensure that the system is functioning according to the requirements and specifications laid down.

CHAPTER 7: SYSTEMS EVALUATION

7.1 Introduction

As the final phase in the life cycle approach, this chapter evaluates the system to identify its strengths and constraints. Some suggestions for future enhancements are also explored.

7.2 Problems Encountered and Solution

Throughout the development process of the system, there were many problems encountered. In this section, the problems encountered will be discussed, and if there are solutions to those problems, the solution would then be stated.

7.2.1 Lack of Wap Development Experience

Lack of experience and knowledge in developing Wap applications has proved to be an obstacle in the beginning. The problem lies with the struggle to understand the concepts of Wap programming and application and differentiate them from the conventional programming concepts to which is more accustomed.

Solution: Luckily, there are abundant reference materials available for the subject, especially from the Internet. To handle this problem, there is no subsitute to hard work, perseverance and advice from coursemates. After the knowledge and skills has been familiarized, everything went on rather smoothly. Most of the time were spent in the Internet, looking through the FAQs to solve some WAP development problems.

7.2.2 Difficulties in Choosing Programming Language and Tools

Choosing the right messaging Application Programming Interface (API) was a problem, as there were many APIs available for the development of a mail system. Furthermore, choosing an API determines the programming language that is to be used.

Problems were again encountered when choosing which server to deploy the applications. As the system were to be developed in Java Servlets, a web server that understands servlets will have to be used.

The wide range of tools often overwhelms a novice who would have problems choosing the right tools for his needs.

Solution: Compare facts, get opinions from experienced people, search the Internet for information and choose the right tools

7.2.3 Lack of Knowledge in Setting Up of Tools

Initially, setting up the Tomcat 3.2.3 server and Java Environment were a problem, as there were many configurations that need to be done in order to get working. Setting up `TOMCAT_HOME`, `JAVA_HOME`, `paths` and `classpaths` were at first a difficult task as a simple addition, deletion, or misplacement of a symbol in the autoexec.bat file, would prevent the tools and environment to work properly.

Solution: Looked for information from the Internet, asked for guidance

7.2.4 Inexperience In Programming Language

Programming Java Servlets and using JavaMail require the knowledge and skills in Java programming. There were quite limited time to master the language.

Solution: Learnt through a lot of examples.

7.2.5 Difficulties in Sending Attachments

Sending attachments involves POSTing a form with encoding type of '`multipart/form-data`'. The servlet must process the POST data in order to extract the encoded file. Unfortunately, there are no methods in the servlets to do so.

Solution: Looked through the JavaMail API FAQs, found out that servlet has no methods for dealing with forms with encoding type of '`multipart/form-data`'. Also, in order for servlets to process these forms, a parser will have to be used. Finally, the 'MultipartRequest Parser' of the O'Reilly package was used.

7.2.6 Difficulties in Reading Attachments

Attachments come in all types. Not all attachments can be read by the browser. Some need to be saved. The problem was to save the attachment file to the server.

Solution: created a method to save the file in the server.

7.2.7 WAP Device Unable to Read 'text/html' Messages

As WML is the markup language of WAP, the WAP device is unable to display messages that are coded in HTML. Initially, all HTML messages could not be read.

Solution: Created a method to sort the HTML 'tags', and to later output the text as strings. All messages that are instances of string, 'text/plain' or 'text/html', were then able to be read.

7.2.8 WAP Device Unable to Display Large Data

WAP browsers has limited capacity to display in that data that are more than 1048 bytes could not be displayed. As a result, the simulator kept displaying an error message, but, the error message did not state that the cause is the data size. The error message only states that 'content could not be encoded by the gateway'.

Solution: Controlled the message to be displayed by limiting only messages that are less than 1048 bytes to be displayed. If message size is more than 1048 bytes, a notification would be displayed to inform users that browser could not display data.

7.3 Evaluation by End Users

7.3.1 Survey Forms Evaluation

Evaluation forms were prepared and distributed to experienced IT people in the faculty, to evaluate the system. Some questions about the performance of the system, the usefulness of the system, and interface of the system were asked. *Refer Appendix B.*

7.4 System Strengths

After months of analysis, design, development and debugging, the WAPES application has several strengths, as follows.

7.4.1 User-Friendly Interface

The interface of WAPES application is simple and easy to use. For WAPESWeb (web-based interface), users can navigate easily to every part of the system, just by clicking on the hyperlinks and buttons. As for WAPESWap, the interface of the simulator is a look alike to a real handphone. Therefore, it is really easy navigate from one card to another.

7.4.2 Alternative Way to Check E-Mail

Overall, the WAPES application provides an alternative way for users to check e-mails. Users will just have to know their POP3 server name, or just select the already available options, and get connected to the servers, to check their e-mails. This is especially a good solution for mobile phone users, or more specifically, office people, to check their e-mails of their company mail boxes.

7.4.3 Facility Provided to Mobile Users

Users can check e-mails at anytime and everywhere as long as they have a WAP phone.

7.4.4 Flexibility to Type or Choose POP3 Server Name

Besides being able to select POP3 server from the available options, users can also type in their own server name, if it's not already in the list. This has shown that the system is flexible and allow users to check any of their POP3 accounts.

7.4.5 Error-Checking

WAPES also has error checking features for all of its inputs. For example, address syntax is checked before a mail is sent. The required fields of the forms also has validation checking, and only those messages that have their required fields completed are able to be sent.

7.4.6 Ability to Perform all basic E-Mail Operations

Regardless of whether it is WAPESWeb or WAPESWap, the application is able to reply, forward and delete a particular message

7.5 System Constraints

7.5.1 Limitation of Capacity

A wireless device has limited input capacity such as numeric keypads. There are specific keys but not a mouse or any pointing device. Wireless Markup Language works with small display screens and it only fits a few lines of text. It has limited low power CPU, small memory and power constraints. As such, the WAPESWap application is also limited by these constraints.

7.5.2 Speed of Connectivity Slow

This is especially true if user's mailbox is too large. The connection speed to the POP3 server becomes slow especially when viewing the INBOX or reading mails in the INBOX.

7.5.3 Cost of Using WAP High – Not Many People Will Use Application

The cost of using WAP is still high and there are doubts that mobile users will hesitate to use the WAPESWap system.

7.5.4 WAPESWap Unable to Display More Than 1048 bytes of Data

Messages that are more than 1048 bytes are unable to be read by the WAPESWap application.

7.5.5 WAPESWap Unable to Read Attachments

Due to the limitations of the wap device, WAPESWap is unable to read attachments.

7.5.6 Single User

As the session information is held in a static class variable, the WAPES application can only be used by a single user at a single time. It would just require a simple session tracking code to replace the static class variables, to make the application multi-user and scaleable.

7.6 Future Enhancements

There are still many more things that can be done to improve the WAPES application. Some suggestions are listed as follows:

- allow multiple accounts to be read, where multiple POP3 servers could be specified
- use `HttpSession` tracking to allow multi-user access
- to implement stricter address error-checking mechanism
- to list in the INBOX, messages that can be read (<1K), and messages that are unable to be read by the Wap browser - users could know which messages could be read, and which could not be read – save connection time
- create dynamically generated decks so that larged size data could be displayed, by creating hyperlinks that link to more data

7.7 Knowledge and Experience Gained

From the development of this project, I have been able to gain more knowledge about JavaMail, Java Servlets, WML and HTML, not to mention WAP technology. I have learnt a lot about installing, configuring and setting up tools. The experience gained from recoding, debugging and doing corrections, has deepen my knowledge in all of the software that I used for this project.

7.8 Conclusion

The system development life cycle is often described as an on going effort and the phases go in circle. Therefore, the life cycle is not finished when the system is first completed. Instead, the system will continue undergoing evaluation, redesign, maintenance and enhancement. In this way the system will evolve better to fulfill user needs. Few systems are perfect for the first time it is produced; therefore it is important to evaluate the system objectively to analyze its strengths and limitations. The system can then be further improved upon.

CHAPTER 8: CONCLUSION

The WAPES project is designed and developed especially for the use of mobile phone users. The core function of the project is to perform standard e-mail functionality through mobile phones. This project has two types of interfaces, the wap-based interface, and the web-based interface. Overall, the final product has fulfilled the basic functional and non-functional requirements as specified in the design.

However, as common for any project, there is still room for improvements. The interface of mobile banking, though adequate and simple to use, can still use much dressing up to look more attractive.

Although there are a lot of limitations on the wap-based-interface, but as a novice, it's very proud to have developed the system because WAP technology is a new technology which is going to be further explored in the future.

Developing the WAPES project is a very novel and exicting experience. On one hand, the system explores the cutting edge Information Technology and on the other hand, it offers practical experience in designing and developing a system for real usage. The knowledge, skills and experience gained in designing and developing this system will be of great use in the future.

References

[1] Machin, R., Toschi, and M., Pedley, C., *Professional WAP*, Wrox Press, 2000

[2] Jeffrey L. W., Lonnie D. B. and Kevin C. D., *System Analysis & Design Methods*, McGraw-Hill, 2000

[3] Deitel, H.M., Deitel P.J. and Nieto, T.R., *Internet and World Wide Web How To Program*, Prentice Hall, 2000

[4] Schneider, G.P. and Perry, J.T., *Electronic Commerce*, Course Technology, 2000

[5] YOURWAP (http://www.yourwap.com, Accessed on August 21, 2001)

[6] myMail (http://www.mymail.ro, Accessed on August 21, 2001)

[7] All about Wap (http://allwap.com, Accessed on July 20, 2001)

[8] How Stuff Works (http://www.howstuffworks.com, Accessed on August 8, 2001)

[9] Wireless in a Nutshell (http://www.wirelessinanutshell.com, Accessed on August 22, 2001)

[10] Nokia Forum (http://www.forum.nokia.com, Accessed on August 15, 2001)

[11] AnywhereYouGo (http://www.anywhereyougo.com, Accessed on July 20, 2001)

[12] VirtualZone (http://virtualzone.de/smartmail/, Accessed on July 21, 2001)

[13] WapUSeek (http://www.wapuseek.com, Accessed on July 19, 2001)

[14] Mailand News (http://www.mailandnews.com, Accessed on July 19, 2001)

[15] E-Mail and Internet Communication (http://supportnet.merit.edu/m-comint/t-interna/text/local.html, Accessed on August 15, 2001)

[16] The Truth About Servers (http://serverwatch.internet.com/articles/servertruths/index.html, Accessed on August 23, 2001)

[17] 1CIS.com (http://www.1cis.com, Accessed on August 23, 2001)

[18] Epinions (http://www.epinions.com, Accessed on August 23, 2001)

[19] R.S Pressman, *Software Engineering: A Practitioners Approach*, McGraw-Hill, 1992.

[20] S.L Pfleeger, Software Engineering: Theory and Practice, 2nd Ed, Prentice Hall, 2001

[21] Java Official Website (http://java.sun.com, Accessed on December 20, 2001)

[22] O'Reilly Website (http://www.servlets.com/cos/index.html, Accessed on January 2, 2001)

APPENDIX A: User Manual

Table of Contents

List of Figures

User Manual

Introduction

Wap E-mail System is mainly a mobile application that lets users check e-mails from any POP3 mail servers, using their mobile phones. The system has two interfaces, wap-based-interface and web-based-interface. It is a simple system with easy-to-use interface.

Objectives

The main objectives of Wap E-Mail System:

- To enable Wap phone users to perform standard e-mail functionality through mobile phones;
- To enable Wap and Web users to check e-mails from any POP3 mail servers

About This Manual

This user manual gives the users step-by-step guide through the application, from setting up to using core functions:

- Hardware and Software Requirements
- Installation and Configuration Guide
- User Guide

Part 1: Hardware and Software Requirements

System Requirements

For the application to function properly, the following requirements are recommended:

Hardware:

Intel Pentium Processor or compatible and above

64 MB RAM or above

Hard Disk Space of At Least 50MB of Free Space

Network Interface Card with Modem/ISDN/LAN/WAN to connect to Internet

Software:

Windows 95/98, Windows NT 4.0 or Windows 2000

Web Server that supports Java (Tomcat Server, Apache JServ, IPlanet)

Nokia Mobile Internet Toolkit

Internet Explorer 5.0

Java Development Kit (JDK)-1.2

Javamail-1.2

Java Activation Framework (JAF)-1.0.1

Pop3-1.1.1

Part 2: Installation and Configuration Guide

2.1 Server Manual

2.1.1 Tomcat 3.2.3 Web Server Installation Guide

1. Download the 'jakarta-tomcat-3.2.3.zip' file from the Internet
2. Extract the zip file into `c:/` drive.
3. A folder called '`jakarta-tomcat-3.2.3`' with tomcat in it would be created
4. To set for Windows9x operating systems:
   a. go to '`c:/autoexec.bat`' file, right-click on the file, choose *Edit*, and type the following in the notepad:

      ```
      set TOMCAT_HOME=c:/jakarta-tomcat-3.2.3
      set JAVA_HOME=c:/jdk1.3.1
      ```

   b. go to the '`c:/config.sys`' file, right-click on the file, choose *Edit*, and type the following in the notepad:

      ```
      shell=\command.com /E:2048/P
      ```

5. To set for Windows 2000 operating systems:
   a. go to *Start->Settings->Control Panel->System*
   b. click on the *Advanced* tab, click on *Environment Variables*
   c. in the lower section titled *System Variables*, click on *New*
   d. enter the following:

      Variable name = **`TOMCAT_HOME`**

      Variable value = **`c:/jakarta-tomcat-3.2.3`**

   e. press 'OK'
   f. again, click on *New*, enter the following:

      Variable name = **`JAVA_HOME`**

      Variable value = **`c:/jdk1.3.1`**

   g. press 'OK'
6. Open a command prompt (MS-DOS) window, change directory to '`c:\jakarta-tomcat-3.2.3\bin`', then, type the command '`tomcat run`'

7. A new .conf file called '`tomcat-apache.conf`' will be setup in '`c:/jakarta-tomcat-3.2.3/conf`'
8. In order to create another directory in the server, go to '`c:/jakarta-tomcat-3.2.3/conf/server.xml`', and the following is added to the context path:

```
<context path="/wapes">
    docbase="<install_dir>:/wapesdev"
    ...
    ...
</context>
```

9. This is to enable that wapes applications to be put in a separate directory in the `wapesdev` folder, instead of the default `webapps` folder
10. As so, the URL to the WAPES applications will be http://localhost:8080/wapes/servlet/

2.1.2 Java Environment Installation Guide

1. Download the following Java products from Java's official website (http://java.sun.com):
   a. JDK1.3.1
   b. Java Activation Framework JAF-1.0.1
   c. JavaMail-1.3.1
   d. Pop3-1.1.1
2. Set the paths and classpaths
3. For Windows9x operating systems:
   a. go to the '`c:/autoexec.bat`' file, right-click on the file, choose *Edit*, and type the following in the notepad:

```
set path=c:\jdk1.3.1\bin;c:\javasoft\jre\1.3.1\bin;
set classpath=c:\jaf-1.0.1\activation.jar;c:\pop3-
1.1.1\pop3.jar;c:\javamail-1.1.3\mail.jar
```

4. For Windows 2000 operating systems:
   b. go to *Start->Settings->Control Panel->System*
   c. click on the *Advanced* tab, click on *Environment Variables*

d. in the lower section titled *System Variables*, click on *New*

e. enter the following:

Variable name = **`path`**

Variable value = *refer no.3*

f. press 'OK'

g. again, click on *New*, enter the following:

Variable name = **`classpath`**

Variable value = *refer no.3*

h. press 'OK'

2.1.3 Setting up File Structure

a. The WAPES source codes are stored into the folder: /wapesdev.

b. Extract the `wapesdev.zip` file from the accompanying disc into installation drive.

2.2 Client Manual

2.2.1 Nokia Mobile Internet Toolkit Installation Guide

1. Download the Nokia Mobile Internet Toolkit from the Internet.
2. Double-click on the downloaded file.
3. The Install Shield Wizard would be launched.
4. Follow the installation instructions that are in the Wizard.

Part 3: User Guide

3.1 Web Module

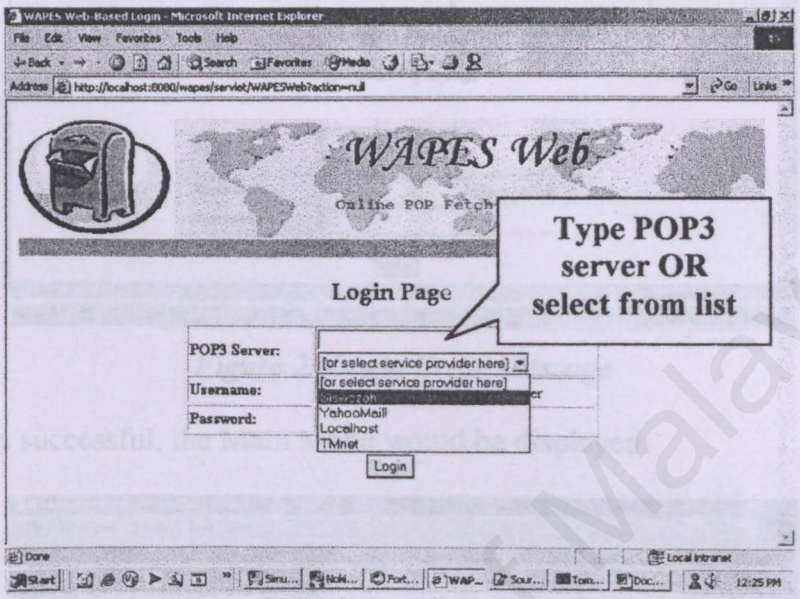


Figure 1: Web Interface Login Page

1. Either type in or select the POP3 server to connect.
2. Type username and password of the mail account of the POP3 server.
3. If username or password is incorrect, or login failure, an error message would be returned.

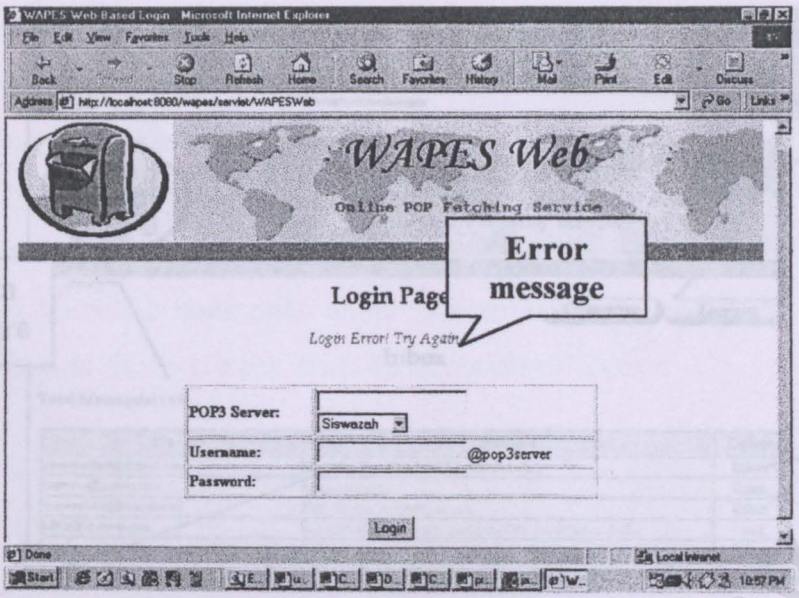


Figure 2: Login Error Message

4. If login is successful, the Main Menu would be displayed.

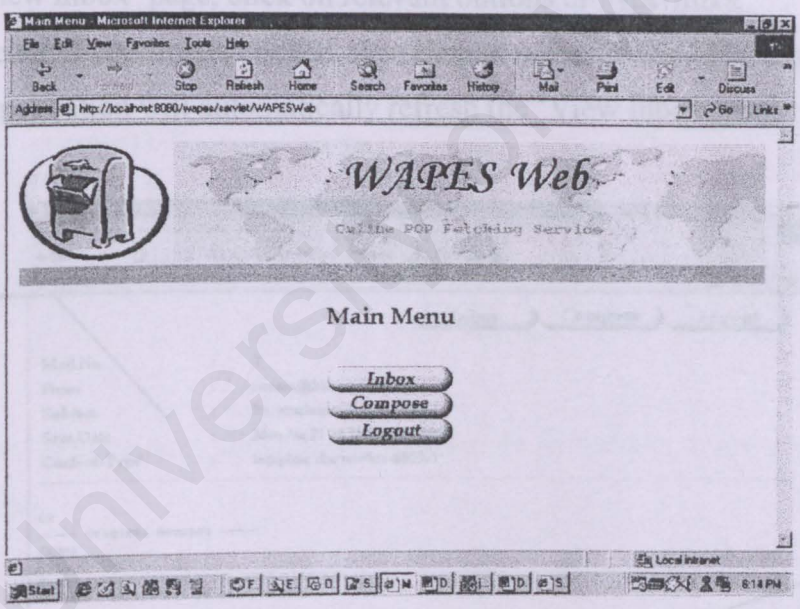


Figure 3: Main Menu

5. In the Main Menu,
   a. To look at the messages in the Inbox, press the **'Inbox'** button
   b. To compose a new mail message, press the **'Compose'** button
   c. To logout of the system, press the **'Logout'** button

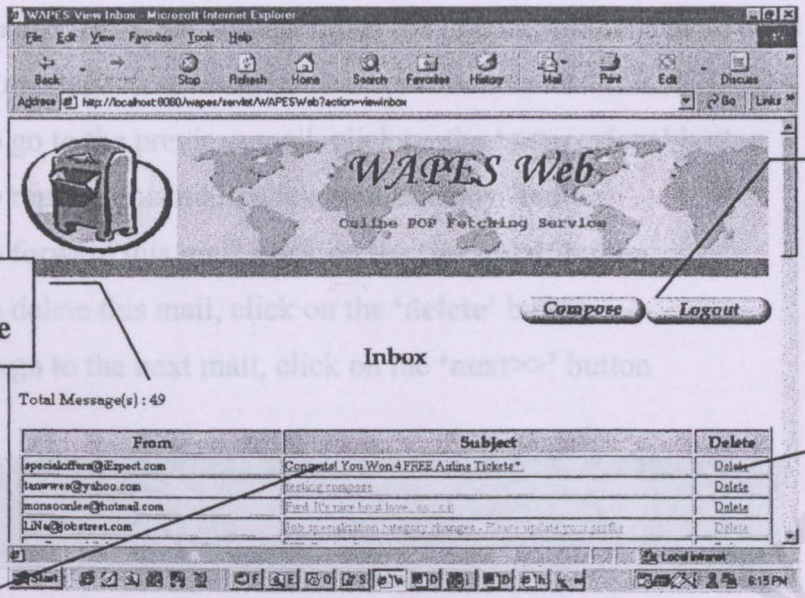


Figure 4: Inbox View

6. In the 'View Inbox' page, click on relevant buttons or hyperlinks.
7. If the 'Delete' hyperlink is clicked, the relevant message would be instantly deleted, and the application will automatically refresh the 'View Inbox' page.

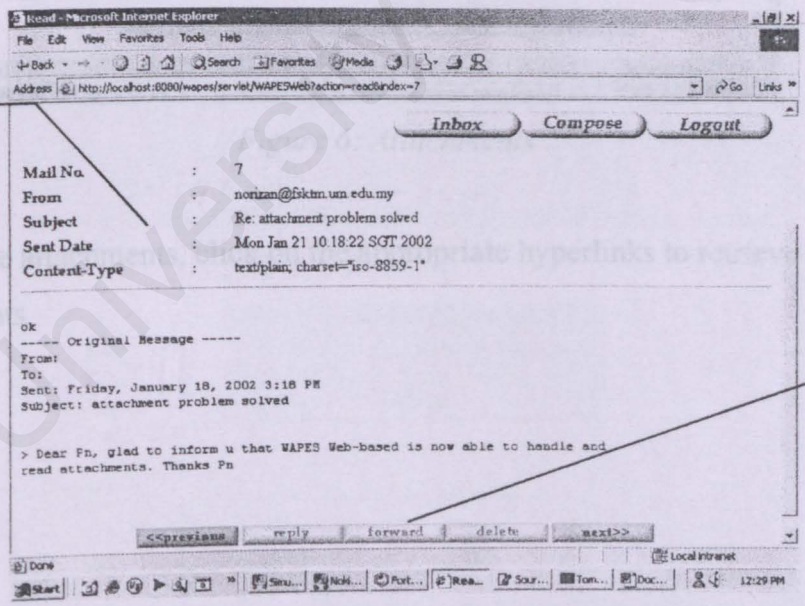
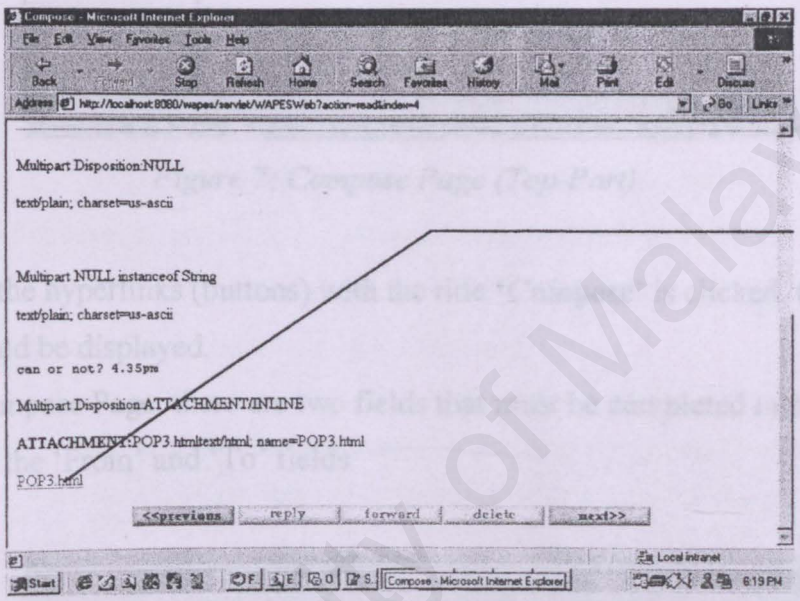


Figure 5: Read Message Page

8. In the 'Read Message Page', the **Message Header** will state the message's description and information.

9. The **Message Operation Bar** specifies the operations that can be done on the particular message
    a. To go to the previous mail, click on the **'<<previous'** button
    b. To reply to this mail, click on the **'reply'** button
    c. To forward this mail, click on the **'forward'** button
    d. To delete this mail, click on the **'delete'** button
    e. To go to the next mail, click on the **'next>>'** button



Click on the hyperlink to retrieve

Figure 6: Attachments

10. If there are attachments, click on the appropriate hyperlinks to retrieve the attachments.

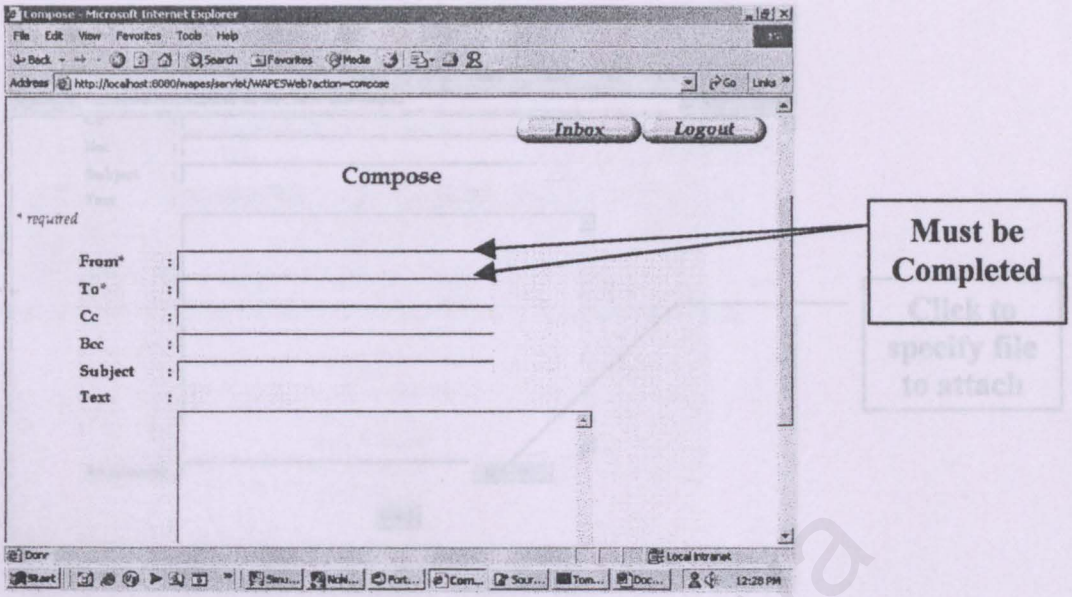


Figure 7: Compose Page (Top-Part)

11. If any of the hyperlinks (buttons) with the title **'Compose'** is clicked, the Compose Page would be displayed.
12. In the Compose Page, there are two fields that must be completed in order to send a message, the 'From' and 'To' fields.

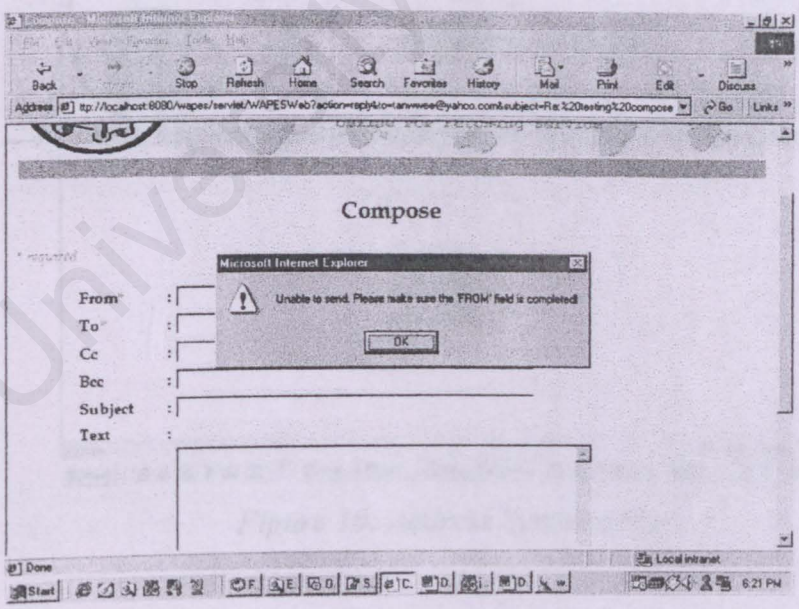


Figure 8: Error Alert

13. If these fields are not completed, a JavaScript alert would be displayed.

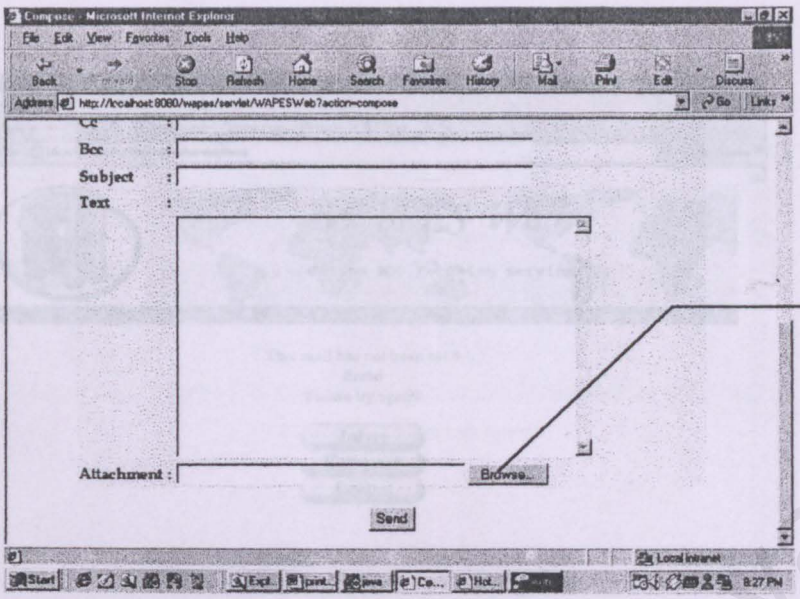


Figure 9: Compose Page (Bottom-Part)

14. To send attachments, click on the 'Browse' button and locate the file that is to be attached.
15. If no atttachments are to be made, leave the input empty.

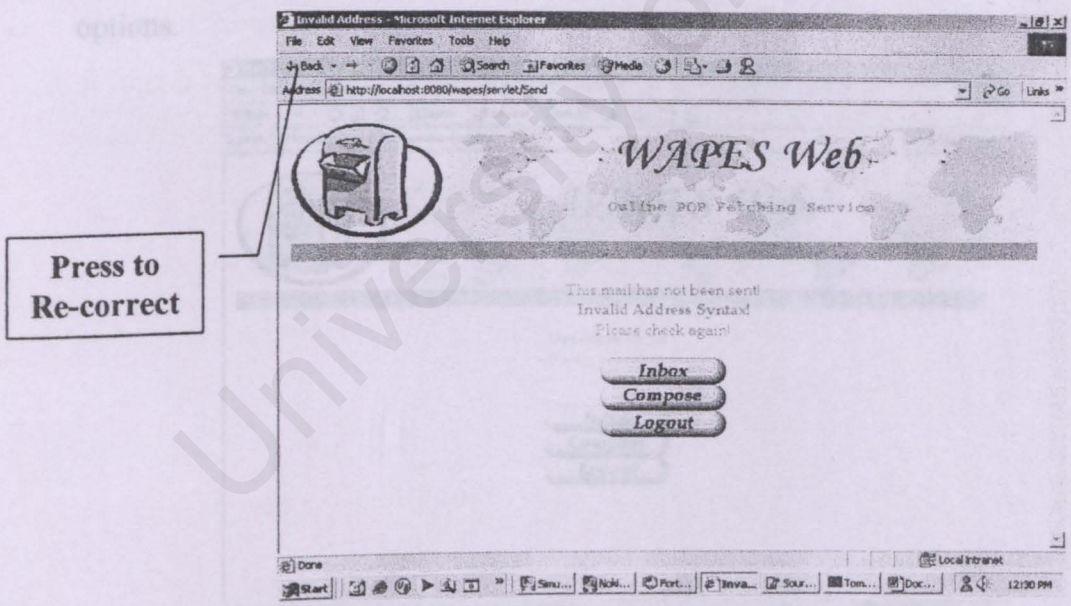


Figure 10: Address Syntax Error

16. If there are address syntax errors, the mail will not be sent.
17. Press the browser's **'Back'** button to do corrections, or proceed by clicking on any of the **Main Menu** options.

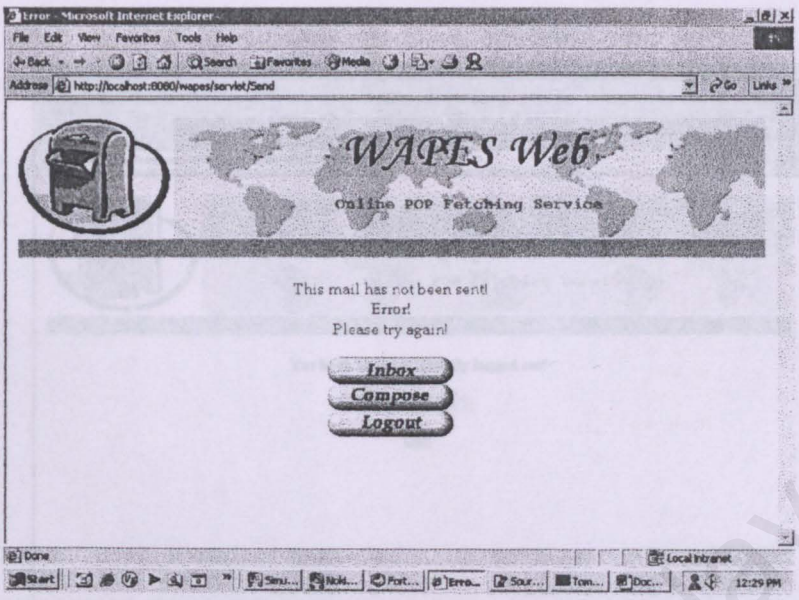


Figure 11: Error Message

18. If an error occurs, an error message would be displayed.
19. Press the browser's **'Back'** button to try again, or proceed with the **Main Menu** options.

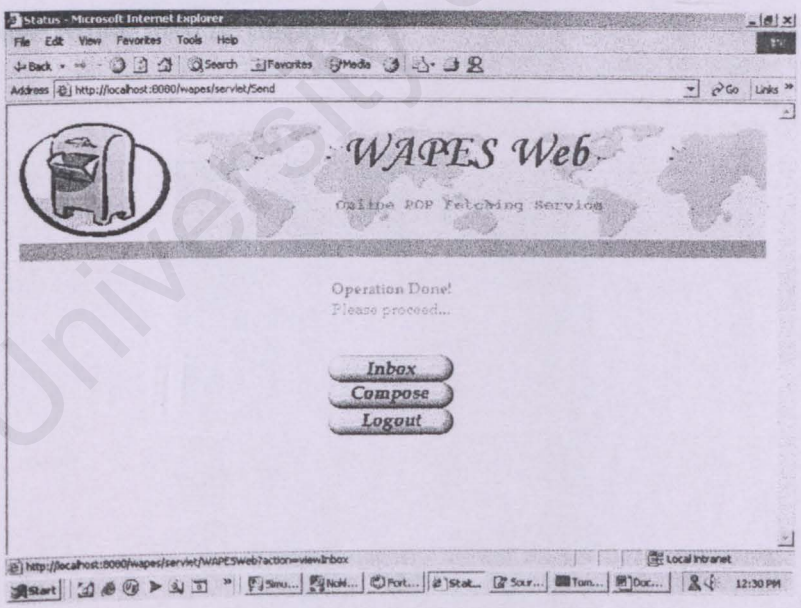


Figure 12: Operation Done Page

20. If sending is successful, the system will display that the operation is done.
21. Proceed with the **Main Menu** options.

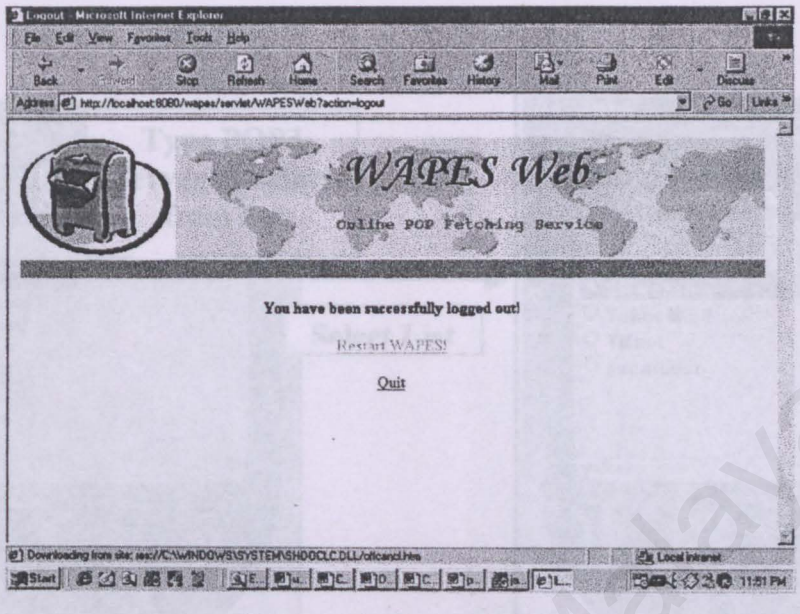


Figure 13: Logout Page

22. To logout, press on any hyperlink (buttons) that has the label name **'Logout'** to logout of the system.
23. A logout page would then be displayed.

3.2 WAP Module

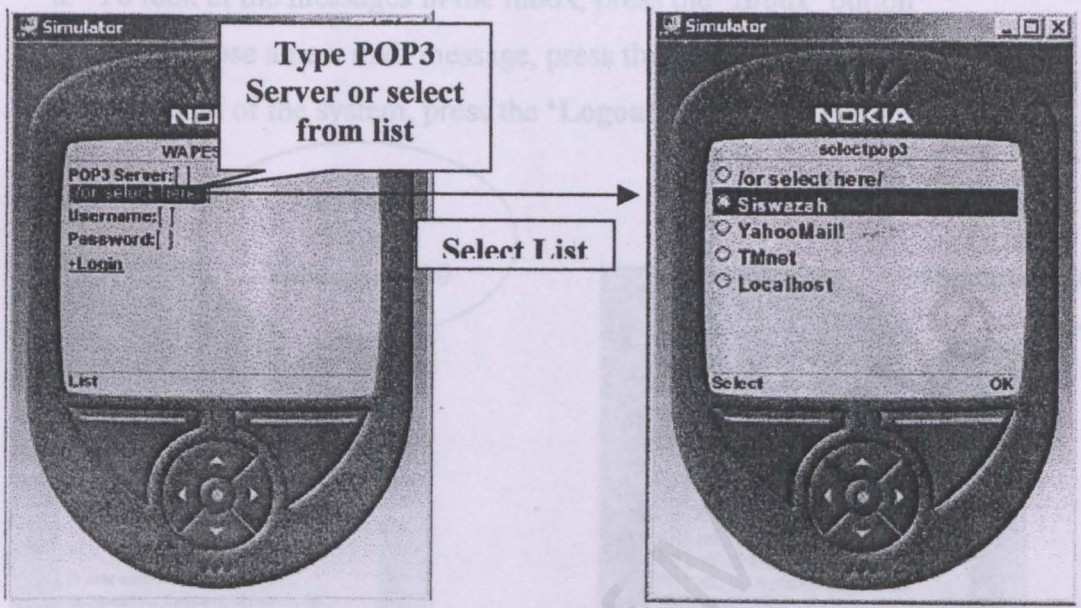


Figure 14: Login Card

1. Either type in or select the POP3 server to connect.
2. Type username and password of the mail account of the POP3 server.

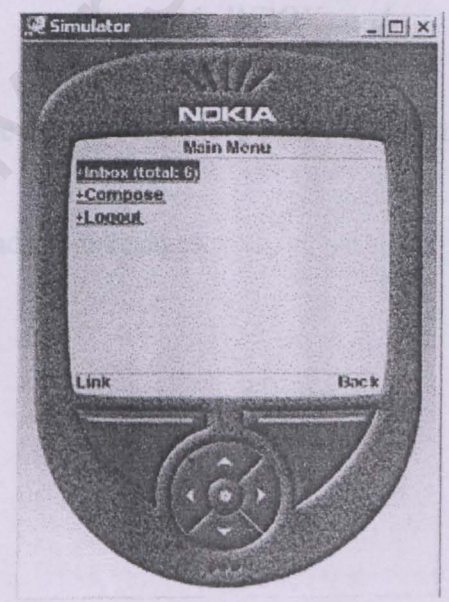


Figure 15: Main Menu

3. If login is successful, the Main Menu would be displayed.
4. In the Main Menu,
    a. To look at the messages in the Inbox, press the **'Inbox'** button
    b. To compose a new mail message, press the **'Compose'** button
    c. To logout of the system, press the **'Logout'** button

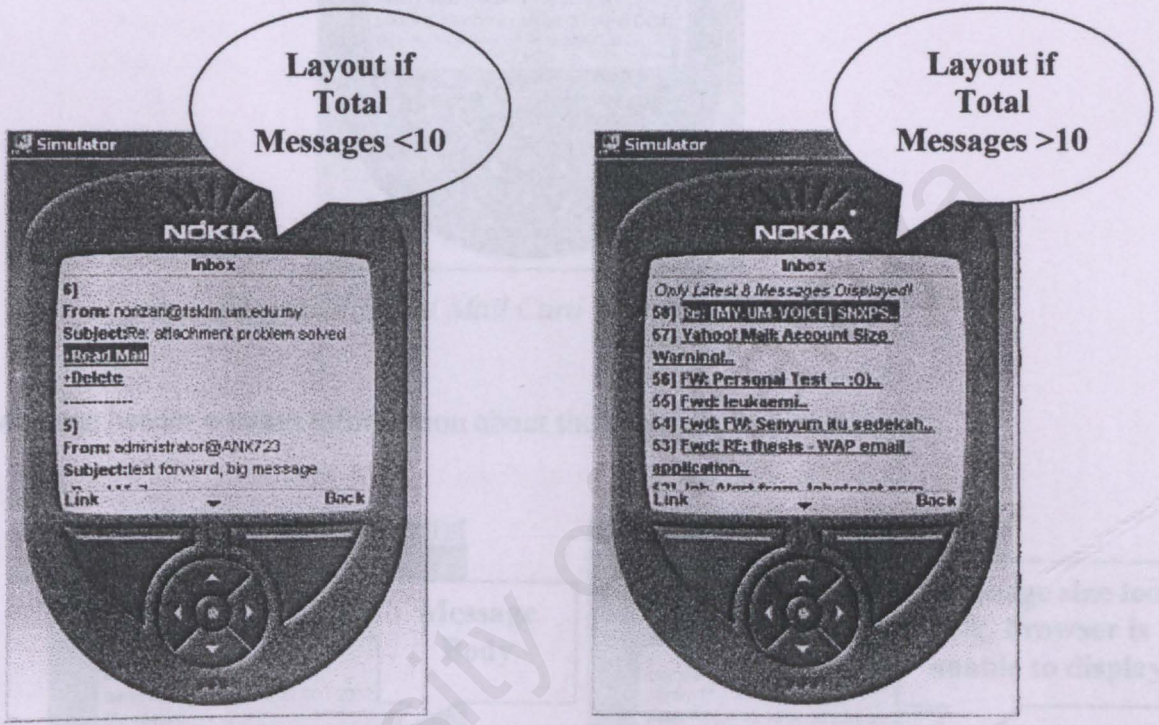


Figure 16: INBOX card

5. If total messages in the INBOX are more than 10, only the latest 8 messages would be displayed.
6. Follow the links to read the messages.

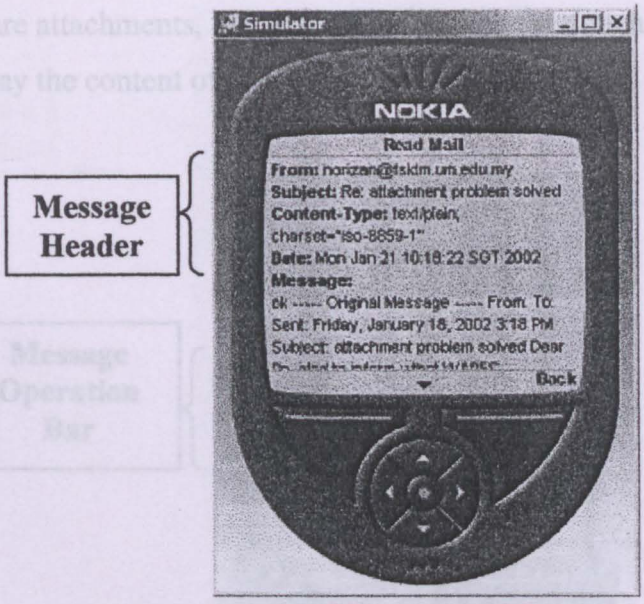


Figure 17: Read Mail Card – Message Header

7. Message header contain information about the particular message.

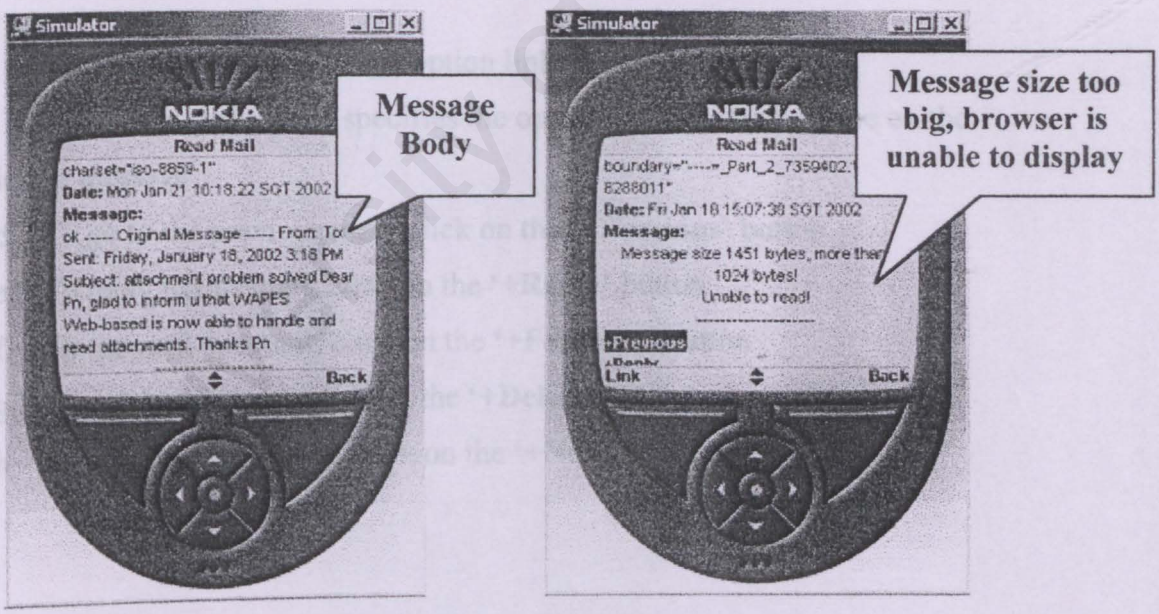


Figure 18: Read Mail Card – Message Content

8. Only messages that are less than 1024 bytes could be displayed in the browser.
9. If message is more than 1024 bytes, the system will return a message that the browser is unable to display.

10. If there are attachments, the system will list out the attachment file names, but, will not display the content of the attachments.

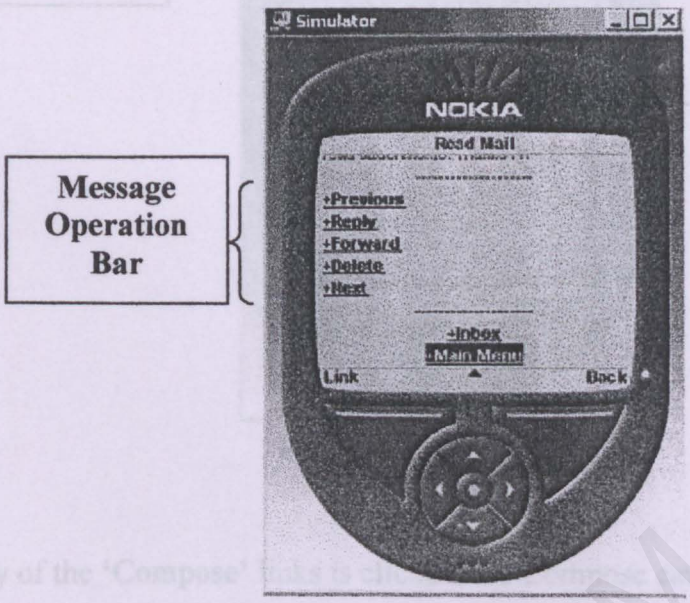


Figure19: Read Mail Card- Message Footer

11. The message footer lists out all the option links.
12. The **Message Operation Bar** specifies the operations that can be done on the particular message
    d. To go to the previous mail, click on the **'+Previous'** button
    e. To reply to this mail, click on the **'+Reply'** button
    f. To forward this mail, click on the **'+Forward'** button
    g. To delete this mail, click on the **'+Delete'** button
    h. To go to the next mail, click on the **'+Next>>'** button

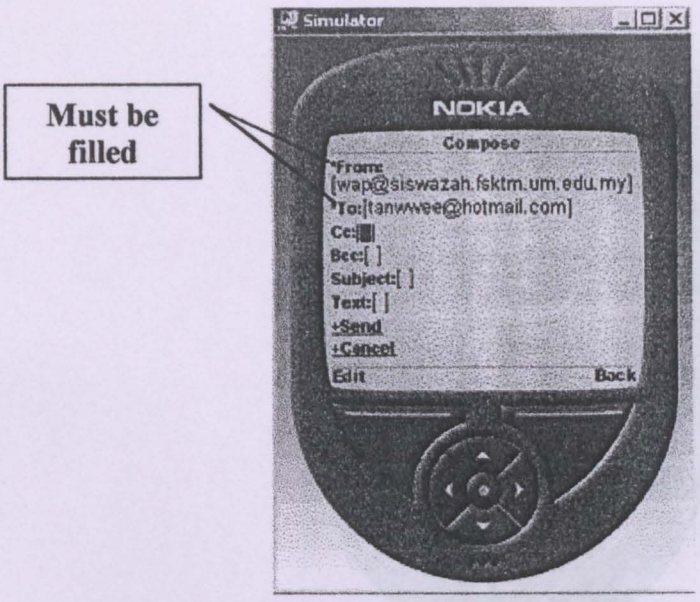


Figure 20: Compose Card

13. If any of the **'Compose'** links is clicked, the Compose card would be displayed.
14. In the Compose card, there are two fields that must be completed in order to send a message, the 'From' and 'To' fields.

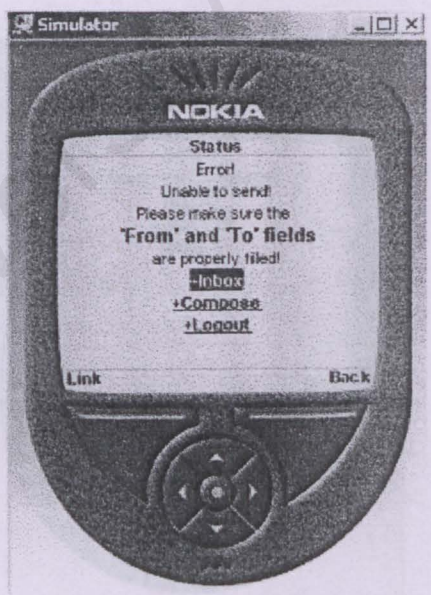


Figure 21: Error Message

15. If these fields are not completed, an error message will be displayed. Press on the '**Back**' link to re-input the required fields.

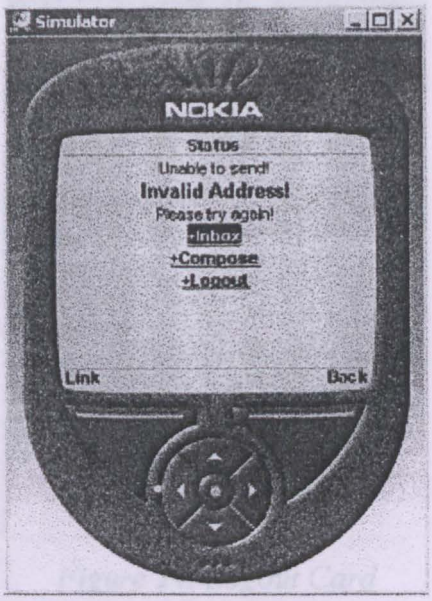


Figure 22: Invalid Address Syntax

16. If there are invalid address syntax in any of the address fields, an error message would be displayed.
17. Press the '**Back**' link to check again.

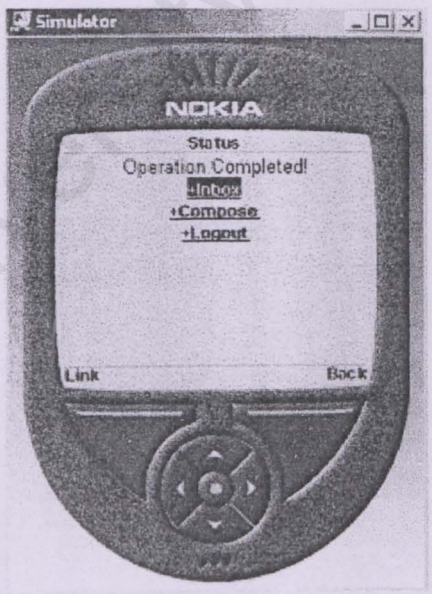


Figure 23: Operation Complete

18. If sending is successful, the system will display that the operation is completed.

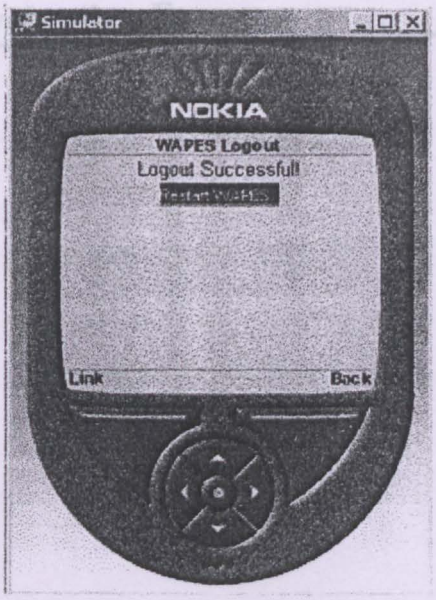


Figure 24: Logout Card

19. To logout, press on any '**Logout**' link.
20. The 'Logout' card would be displayed.

APPENDIX B: System Evaluation Forms

WAP E-Mail System (WAPES)
Final Year Project
Faculty of Computer Science & Information Technology

System Evaluation Form

1. How do you find the performance of the system?
   ☑ Excellent ☐ Good ☐ Fair
2. Do you think the system is useful for mobile phone users?
   ☑ Yes ☐ No
3. If you have a WAP phone, would you use this system?
   ☑ Yes ☐ No
4. How does the links in the system facilitate your navigation?
   ☐ Very Good ☑ Good ☐ Fair
5. Is the system interface user-friendly?
   ☑ Yes ☐ Not Quite ☐ No
6. Are the system instructions clear?
   ☑ Yes ☐ No
7. Is the system response time acceptable?
   ☑ Fast ☐ Not So Fast ☐ Acceptable
8. Are you satisfied with the security features?
   ☑ Yes ☐ No
9. Do you intend to introduce the system for use by your friends?
   ☑ Yes ☐ No
10. In your opinion, what are the future enhancements needed for the system?

    _____

Name: KOH YUN SING
Signature: _____
Occupation: TUTOR

WAP E-Mail System (WAPES)
Final Year Project
Faculty of Computer Science & Information Technology

System Evaluation Form

1. How do you find the performance of the system?

   ☐ Excellent ☑ Good ☐ Fair

2. Do you think the system is useful for mobile phone users?

   ☐ Yes ☑ No

3. If you have a WAP phone, would you use this system?

   ☐ Yes ☑ No

4. How does the links in the system facilitate your navigation?

   ☐ Very Good ☑ Good ☐ Fair

5. Is the system interface user-friendly?

   ☐ Yes ☑ Not Quite ☐ No

6. Are the system instructions clear?

   ☑ Yes ☐ No

7. Is the system response time acceptable?

   ☐ Fast ☐ Not So Fast ☑ Acceptable

8. Are you satisfied with the security features?

   ☐ Yes ☑ No

9. Do you intend to introduce the system for use by your friends?

   ☐ Yes ☑ No

10. In your opinion, what are the future enhancements needed for the system?

    Add hotmail.

Name: Lai Siew Ching.

Signature: Li

Occupation: Tutor

WAP E-Mail System (WAPES)
Final Year Project
Faculty of Computer Science & Information Technology

System Evaluation Form

1. How do you find the performance of the system?

   ☐ Excellent ☑ Good ☐ Fair

2. Do you think the system is useful for mobile phone users?

   ☑ Yes ☐ No

3. If you have a WAP phone, would you use this system?

   ☑ Yes ☐ No

4. How does the links in the system facilitate your navigation?

   ☑ Very Good ☐ Good ☐ Fair

5. Is the system interface user-friendly?

   ☑ Yes ☐ Not Quite ☐ No

6. Are the system instructions clear?

   ☑ Yes ☐ No

7. Is the system response time acceptable?

   ☐ Fast ☑ Not So Fast ☐ Acceptable

8. Are you satisfied with the security features?

   ☑ Yes ☐ No

9. Do you intend to introduce the system for use by your friends?

   ☑ Yes ☐ No

10. In your opinion, what are the future enhancements needed for the system?

    better interface

Name: WONG SHIN YI

Signature: [signature]

Occupation: student